



## Homematic IP noch vielseitiger steuerbar Homematic IP RFUSB Stick für alternative Steuerungsplattformen

Infos zum Bausatz

im ELV Shop

#10145

Mit dem Bausatz ELV HM-MOD-RPI-PCB [1] kann man einen Raspberry PI als Homematic IP/Homematic Zentrale nutzen. Der hier vorgestellte Bausatz HmIP-RFUSB erweitert die Möglichkeiten nochmals erheblich, das Homematic IP Kommunikationsprotokoll auf anderen Hardwareplattformen über einen einfachen Standard-USB-Anschluss zu nutzen. So stehen fertige Lösungen auf Basis des RaspberryMatic Projekts zur Verfügung, aber auch völlig neue Lösungen, die die Virtualisierungs-Technologie „Docker“ verwenden. Damit lässt sich die CCU2-Software auf weiteren Consumer-Produkten wie z. B. NAS-Stationen nutzen, um eine Haussteuerung mit Homematic IP Komponenten aufbauen zu können.

### Homematic IP für Entwickler

Für die Systeme Homematic und Homematic IP existieren neben den von eQ-3 bzw. deren Handelspartnern angebotenen Standard-Hardware-Zentralen zahl-

reiche weitere Hard- und Softwarelösungen wie zum Beispiel RaspberryMatic, bei denen ein Raspberry Pi als Hardwarebasis dient und die entsprechend entwickelten Linux-Distributionen als eigenständige Distribution arbeiten.

Der hier vorgestellte Homematic IP RFUSB Stick dient mit seinem Funktionsumfang einmal als Entwicklungswerkzeug, das Softwareentwicklern unter Nutzung des Software-Development-Kits (SDK) erlaubt, von eben diesen, auf einem PC laufenden Entwicklungswerkzeugen aus mit Homematic IP Komponenten zu kommunizieren.

Andererseits erlaubt der HmIP-RFUSB auch die Nutzung alternativer Hardwareplattformen wie eben RaspberryMatic oder weiterer Rechnerplattformen durch den interessierten Nutzer, sodass dieser weder auf die CCU2 noch auf die allein für den Raspberry Pi geschaffene Interface-Lösung HM-MOD-RPI-PCB angewiesen ist.

Technische Daten

Geräte-Kurzbezeichnung:	HmIP-RFUSB
Spannungsversorgung:	5 Vdc (USB powered)
Schutzart:	IP20
Umgebungstemperatur:	5–35 °C
Funk-Frequenzband:	868,0–868,6 MHz 869,4–869,65 MHz
Max. Funk-Sendeleistung:	10 dBm
Empfängerkategorie:	SRD category 2
Typ. Funk-Freifeldreichweite:	300 m
Duty Cycle:	< 1 % pro h / < 10 % pro h
Abmessungen (B x H x T):	28 x 84 x 11,5 mm (ohne Stecker)
Gewicht:	18 g



Alle notwendigen Tools, Treiber für das eingesetzte Betriebssystem und Programme, um den HmIP-RFUSB Stick verwenden zu können, stehen im OCCU-Github-Repository [2] zur Verfügung und ermöglichen es, den HmIP-RFUSB Stick in beliebigen eigenen Softwareprojekten einsetzen zu können.

### Betriebssystemvoraussetzungen

Für Windows steht ein Treiber zur Verfügung, der über [3] heruntergeladen werden kann. Unter Linux muss der cp210x-Treiber installiert sein, der über folgende Befehle geladen und konfiguriert werden muss:

```
modprobe cp210x
sh -c 'echo 1b1f c020 > /sys/bus/usb-serial/drivers/cp210x/new_id'
```

Mit der ersten Zeile wird der Treiber geladen, die zweite Zeile sorgt dafür, dass der Treiber die eQ-3-Vendor-ID (1b1f) und Product-ID (c020) berücksichtigt. Nach der Konfiguration steht ein Device z. B. `/dev/ttyUSB0` zur Verfügung.

### Softwarekomponenten im OCCU-Projekt

Das OCCU-Projekt stellt verschiedene Softwarekomponenten für Homematic IP/Homematic zur Verfügung, mit der sich unterschiedliche Lösungen realisieren lassen.

Die Basis dazu bilden die Schnittstellenprozesse, um Homematic IP/Homematic Geräte über eine Programmierschnittstelle (z. B. XML-RPC) steuern zu können. Die Schnittstellenprozesse kommunizieren auf der einen Seite mit der Funkhardware (HmIP-RFUSB, HM-MOD-RPI-PCB), auf der anderen Seite stellen Sie eine Programmierschnittstelle zur Verfügung.

Für den neuen HmIP-RFUSB Stick stehen folgende Programme zur Verfügung:

- Java Tool, um administrative Aufgaben wie Firmware-Update durchführen zu können

- Java HmIP Schnittstellenprozess crRFD (analog rfd bei Homematic)
- Speziell für die CCU-Software ein HmIP Server, der eine Kombination aus dem HM Server (Gruppenverwaltung, Messdatenerfassung) der CCU2 und dem HmIP Schnittstellenprozess ist.

Das Java Tool besteht aus einem einzigen Jar-Archiv „*hmip-copro-update.jar*“.

Ruft man

```
java -jar hmip-copro-update-jar-with-dependencies.jar
```

ohne weitere Parameter auf, wird ein Hilfetext ausgegeben:

```
java -jar hmip-copro-update.jar
This is the Homematic IP Coprocessor Updater. Usage:
java -jar hmip-copro-update.jar -p <serial port> [-f <update file>] [-v]
-p <serial port>: Serial port to which the coprocessor is connected
-f <update file>: Update the coprocessor with the given firmware file
-o : Overwrite the coprocessor firmware even if it already has the
    desired version
-v : Print current coprocessor version
-r : Perform coprocessor factory reset
```

Um zu überprüfen, ob der HmIP-RFUSB Stick richtig funktioniert, kann zum Beispiel die Versionsnummer abgefragt werden:

```
java -jar hmip-copro-update.jar -p /dev/ttyUSB0 -v
[INFO] Checking coprocessor firmware version ...
[INFO] Bootloader version = 1.0.11
[INFO] Application version = 2.4.4
```

Um ein Firmware-Update (Tabelle 1) einzuspielen, muss das Tool wie folgt aufgerufen werden:

```
java -jar hmip-copro-update-jar-with-dependencies.jar -p /dev/
ttyUSB0 -f ./HMIP-RFUSB-2.8.4.eq3
Performing update with firmware file ./home/user/./
HMIP-RFUSB-2.8.4.eq3' ...
[INFO] Updating coprocessor from version 2.4.4 to version 2.8.4
[INFO] Entering bootloader ...
[INFO] Bootloader version: 1.0.11
[INFO] Starting application ...
[INFO] New application Version: 2.8.4
```

Eine weitere Softwarekomponente ist der crRFD. Hierbei handelt es sich um den Homematic IP Schnittstellenprozess analog dem rfd für Homematic.

### Funk-Hardware/Programme

#### 1.) HM-MOD-RPI-PCB: sowohl Homematic IP wie auch Homematic Funk

- Firmware für das Funkmodul: `dualcopro_si1002_update_blhm.eq3`
- multimacd-Programm einschl. spezieller Kernelmodule
- Homematic IP Schnittstellenprozess crRFD
- Homematic Schnittstellenprozess rfd

#### 2.) HM-MOD-RPI-PCB: nur Homematic Funk

- Firmware für das Funkmodul: `coprocessor_update.eq3`
- Homematic Schnittstellenprozess rfd

#### 3.) HmIP-RFUSB Stick: nur Homematic IP Funk

- Firmware für den USB-Stick: `HMIP-RFUSB-2.8.4.eq3`
- Homematic IP Schnittstellenprozess crRFD

#### 4.) HmIP-RFUSB Stick und LAN-Gateway (HM-LGW-0-TW-W-EU)

- Firmware für den USB-Stick: `HMIP-RFUSB-2.8.4.eq3`
- Homematic IP Schnittstellenprozess crRFD (USB-Stick)
- Homematic Schnittstellenprozess rfd (LAN-Gateway)

Der crRFD wird über eine Konfigurationsdatei konfiguriert. Eine Liste der Konfigurationsparameter einschließlich Funktionsbeschreibung und Standardwerte sowie eine Beispielkonfigurationsdatei befindet sich im OCCU-Repository [2].

Der crRFD wird wie folgt gestartet:

```
java -jar crRFD-local-with-
devices-1.10.0-SNAPSHOT.jar
./crRFD.conf
```

Einen Überblick, welche Programme man für welche Hardware benötigt, liefert die Übersicht in Tabelle 1.



Auf Basis der OCCU-Software wurden in den letzten Jahren unterschiedliche Open-Source-Lösungen realisiert. Neben sehr speziellen Lösungen wie zum Beispiel dem Homematic Manager [4] gibt es auch Lösungen, die die Software der CCU2 auf einer anderen Hardwareplattform zur Verfügung stellen. Gerade bei sehr großen Installationen wirkt die Web-Oberfläche der CCU2 träge und lässt sich auf Systemen wie z. B. dem Raspberry Pi 3 flüssiger bedienen.

Um die Software der CCU2 auf einem Raspberry PI zu betreiben gibt es zwei unterschiedliche Ansätze:

- 1.) Auf einem vorhandenen Betriebssystem wie Raspbian wird die CCU2-Software in einem VM-Container (LXC) zur Verfügung gestellt. Dieser Ansatz wird zum Beispiel von YAHM [5] verfolgt. Vorteil dieser Lösung ist, dass neben der CCU2-Software ein Raspberry PI auch andere Aufgabe wie Fileserver übernehmen kann. Nachteil ist eine komplexere Installation, die nur von Anwendern mit entsprechendem Linux-Know-how durchgeführt werden kann.
- 2.) RaspberryMatic verfolgt zum Beispiel den Ansatz, das Betriebssystem der CCU2 nachzubauen. Es bleibt aber nicht bei einer Kopie der CCU2-Software, vielmehr werden aktuelle Versionen der Programme und des Linux-Kernels eingesetzt und zusätzliche Erweiterungen wie z. B. die Unterstützung von USV-Anlagen integriert. Nähere Details werden auf der Projektseite des RaspberryMatic Projekts [6] beschrieben.

### RaspberryMatic

Mit Erscheinen dieser Ausgabe des ELV Journal unterstützt die RaspberryMatic neben dem GPIO-basierten Funkmodul [1] ebenfalls den HmIP-RFUSB Stick. Dies ist besonders für Anwender interessant, die die GPIO-Erweiterungsleiste des Raspberry Pi für andere Hardware benötigen. Mit dem HmIP-RFUSB Stick und einem Homematic LAN-Gateway [7] stehen somit die gleichen Funktionen zur Verfügung wie mit einem HM-MOD-RPI-PCB.

Steckt man einen HmIP-RFUSB an einen der USB-Anschlüsse seines unter RaspberryMatic betriebenen Raspberry Pi und startet diesen neu, erkennt das System selbstständig dessen Existenz, indem es, wie weiter oben beschrieben, in der Start-up-Datei:

```
/etc/init.d/S00eQ3SystemStart
```

mittels der folgenden Anweisungen das entsprechende Kernelmodul initialisiert:

```
if lsusb | grep -q 1b1f:c020; then
  modprobe cp210x
  echo 1b1f c020 >/sys/bus/usb-serial/drivers/cp210x/new_id
  if [[ -e /dev/ttyUSB0 ]]; then
    HMIP_DEV="HmIP-RFUSB"
    HMIP_DEVNODE="/dev/ttyUSB0"
  fi
fi
```

Wie zu erkennen ist, wird nach dem Laden des Kernelmoduls eine globale Variable HMIP\_DEV gesetzt, die im späteren Verlauf in eine temporäre Datei `/var/mode` abgespeichert wird, damit bei weiteren Starts anderer Prozesse diese ihr Startverhalten darauf anpassen können.

Ist dieser initiale Schritt erledigt und der HmIP-RFUSB korrekt erkannt und initialisiert, startet RaspberryMatic wie gewohnt die notwendigen Systemprozesse, um eine Kommunikation mit den Homematic IP Geräten zu ermöglichen.

Allen voran sei hier der Java-basierte HMIP Server genannt, der vorher auf das entsprechende `/dev/ttyUSB0` Device mittels Anpassung der `/var/etc/crRFD.conf` Datei umgestellt wurde.

Darüber hinaus unterbindet RaspberryMatic den Start von `multimacd`, eines Dienstes, der lediglich für die gemeinsame Funkkommunikation von Homematic und Homematic IP mit demselben Funkmodul notwendig ist.

Um jedoch weiterhin, auch mit Einsatz eines HmIP RFUSB, Homematic Geräte ansteuern zu können, kann man hierfür entweder ein LAN-

Gateway von eQ-3 (HM-LGW-0-TW-W-EU) oder einen zweiten Raspberry Pi (z. B. auf Basis des kompakten Raspberry Pi Zero) mit RaspberryMatic einsetzen. In diesem Fall regelt RaspberryMatic dann selbstständig, dass Homematic Funkanfragen entsprechend über das LAN-Gateway bzw. Homematic IP Anfragen über den HmIP-RFUSB geleitet werden.

Neben der RaspberryMatic gibt es ein zweites, recht neues Projekt, das auf Basis des HmIP-RFUSB eine alternative Plattform zur CCU2 in Form eines Docker-Containers für X86-Systeme zur Verfügung stellt. Docker ist eine Virtualisierungslösung, die z. B. von einigen NAS-Herstellern unterstützt wird. Die X86-Plattform bietet noch einmal ein Geschwindigkeitsvorteil gegenüber einem Raspberry Pi.

### Docker-Container

Aufgrund der großen Vielfalt beim Einsatz von Docker-Containern und der daraus resultierenden Komplexität kann an dieser Stelle leider nur exemplarisch auf eine Installation eingegangen werden. Weitergehende Hilfestellungen erhält man auf der Projektseite [8] und im Homematic Forum [9].

An dieser Stelle beschreiben wir die Installation unter dem zum Zeitpunkt des Redaktionsschlusses aktuellen Debian-Linux 9.1. Voraussetzung ist, dass Linux installiert und betriebsbereit ist und ein Internetzugang besteht.

Zur Vereinfachung der Anleitung gehen wir davon aus, dass man alle Befehle als root-User ausführt. Wenn man dies nicht macht, ist den Befehlen eventuell ein „`sudo`“ voranzustellen. Installieren Sie zunächst die `usbutils`, um wichtige Verwaltungsprogramme für USB zu installieren:

```
apt-get install usbutils
```

Nach der Installation muss man ein Update der USB-IDs durchführen, damit der USB-Stick korrekt erkannt wird:

```
update-usbids
```

Jetzt ist sicherzustellen, dass das Modul des USB-Treibers `cp210x` für den HmIP-RFUSB Stick ist und der Treiber sich für den Stick verantwortlich fühlt:

```
modprobe cp210x
echo 0x1b1f 0xc020 >/sys/bus/usb-serial/drivers/cp210x/new_id
```

Dann ist der HmIP-RFUSB Stick an den PC anzuschließen und herauszufinden, über welches Device der USB-Stick angesprochen werden kann:

```
dmesg |grep tty
```

In diesem Beispiel ist der USB-Stick an das Device `ttyUSB0` angeschlossen. Dies merkt man sich, wir kommen darauf später zurück.

```
[981363.260109] usb 3-1: cp210x converter now attached to ttyUSB0
```

Jetzt ist die Docker-Community-Edition (CE) für das verwendete Betriebssystem [10] zu installieren. In der sehr guten Dokumentation dazu [11] ist die Installation für das jeweilige Betriebssystem erklärt.



Wenn man Docker auf einem NAS betreiben will, ist die Dokumentation des NAS-Herstellers zurate zu ziehen, zum Beispiel für Synology NAS [12] oder QNAP-NAS [13].

```
apt-get install docker-ce
```

Dann startet man den vorbereiteten Docker-Container. Beim ersten Start wird dieser aus dem „Docker-Hub“ genannten Repository heruntergeladen. Im folgenden Aufruf ist eventuell der Pfad zum oben gemerkten USB-Device auf der linken Seite der Zeile:

```
--device=/dev/ttyUSB0:/dev/ttyUSB0:rwm anzupassen, zum Beispiel:  
--device=/dev/[Ihr_gemerakter_devicename]:/dev/ttyUSB0:rwm
```

```
docker run --name dccu2-x86_64 \  
-p 2222:22 \  
-p 80:80 \  
-p 1900:1900 \  
-p 2001:2001 \  
-p 2002:2002 \  
-p 2010:2010 \  
-p 8181:8181 \  
-p 9292:9292 \  
-e PERSISTENT_DIR=/usr/local \  
-v ccu2_data:/usr/local \  
-v /dev/bus/usb:/dev/bus/usb \  
--device=/dev/ttyUSB0:/dev/ttyUSB0:rwm \  
--hostname homematic-dccu2 \  
-d --restart=always \  
litti/dccu2-x86_64
```

Nach dem Download startet der Container und nach einer kurzen Wartezeit kann man die WebUI durch Eingabe der lokalen IP-Adresse des Hosts im Browser aufrufen.

Wichtig ist, dass die benötigten Ports auf dem lokalen Host (das sind die Angaben mit -p im obigen Befehl auf der linken Seite vom Doppelpunkt) frei sind. Sollte beim Start eine Meldung erscheinen, dass dieser fehlschlägt, weil der entsprechende Port bereits belegt ist, dann sollte man entweder den obigen Befehl anpassen oder den Port lokal frei machen. Wird beispielsweise auf dem lokalen Port 80 am Host bereits ein Web-Server betrieben, kann man den lokalen Port 8585 an den Docker-Container weiterleiten:

```
docker run --name dccu2-x86_64 \  
-p 2222:22 \  
-p 8585:80 \  
-p 1900:1900 \  
-p 2001:2001 \  
-p 2002:2002 \  
-p 2010:2010 \  
-p 8181:8181 \  
-p 9292:9292 \  
-e PERSISTENT_DIR=/usr/local \  
-v ccu2_data:/usr/local \  
-v /dev/bus/usb:/dev/bus/usb \  
--device=/dev/ttyUSB0:/dev/ttyUSB0:rwm \  
--hostname homematic-dccu2 \  
-d --restart=always \  
litti/dccu2-x86_64
```

In diesem Zusammenhang ist zu beachten, dass viele Apps die CCU2-Ports für rfd (2001), hs485d (2002), HMIP Server (2010), regascrip (8181) etc. direkt ansprechen wollen und im Regelfall keine Eingabemöglichkeit für einen abweichenden Port bieten. Sollte einer dieser Ports lokal im Host belegt sein, bleibt nichts anderes übrig, als diesen Port lokal zu räumen.

Weitere hilfreiche Docker-Befehle sind unter anderem:

- Status aller Docker-Container abrufen:

```
docker ps -a
```

- Laufenden dccu2-Container beenden:  
`docker kill $(docker ps | grep „dccu2-x86_64“ | awk ‚{print $1}‘)`
- Über Konsole auf den laufenden dccu2-Container zugreifen (diese kann mit „exit“ wieder verlassen werden):  
`docker exec -i -t $(docker ps | grep „dccu2-x86_64“ | awk ‚{print $1}‘) /bin/bash`

Weitergehende Aktionen wie zum Beispiel die Unterstützung des Back-up-Einspielens und die Umsetzung von Add-on-Installationen sind für den Zeitpunkt nach Redaktionsschluss dieses ELV Journals geplant. Aus diesem Grund kann die konkrete Vorgehensweise zur Einspielung von Back-ups etc. hier noch nicht beschrieben werden. Details dazu werden auf der Projektseite [14] veröffentlicht.

## Schaltung

Die Schaltung des HmIP-RFUSB ist in Bild 1 dargestellt. Die Spannungsversorgung und Kommunikation erfolgt via USB mittels des USB-Steckers BU1. Der Schnittstellenwandler IC1 vom Typ CP2102 übernimmt die gesamte Konvertierung der Datensignale. An Pin 4 (D+) und Pin 5 (D-) erfolgt der Anschluss an den USB-Port. IC-intern erfolgt dann die Umsetzung der differenziell ankommenden Datensignale in UART-Signale, die dann an den entsprechend bezeichneten Ausgängen (Pins 1, 2, 11 und 12 sowie 23–28) zur Verfügung stehen. Das Dioden-Array D2 dient dem Schutz vor elektrostatischer Entladung. Die Reset-Schaltung des ICs ist mit dem Widerstand R3 realisiert.

Die Kondensatoren C7 bis C14 dienen der Stabilisierung und Filterung der Versorgungsspannung. Herzstück der Schaltung ist der Mikrocontroller IC2 vom Typ EFM32F200F64, er arbeitet mit einem intern erzeugten Takt von 14 MHz. Zur Peripherie des Controllers gehört neben den Filter- und Stützkondensatoren die Duo-LED D1 samt den zugehörigen Widerständen R1 und R2, die verschiedene Betriebszustände, z. B. bei der Inbetriebnahme und bei der Anmeldung an die Zentrale oder das Senden an Verknüpfungspartner mit den Farben Rot, Grün und Orange signalisiert.

Die Kommunikation erfolgt schließlich über Funk mittels des Transceiverbausteins TRX1, der ebenfalls vom Controller IC1 überwacht und angesteuert wird. Diesem werden mit C15 und C16 jeweils ein Stütz- und Filterkondensator zur Seite gestellt.

## Nachbau

Der Bausatz wird bis auf wenige Löt- und Montagearbeiten weitgehend vorgefertigt geliefert. Der vollständige Lieferumfang ist in Bild 2 zu sehen.

So beginnt der Aufbau mit einer Sichtkontrolle auf ordnungsgemäße Bestückung und Lötfehler entsprechend den Platinenfotos (Bild 3) und den zugehörigen Bestückungsplänen sowie der Stückliste.

Die Montagearbeiten beginnen mit dem Auflöten des Transceivermoduls TRX1. Dafür wird die Stiftleiste mit den langen Stiften voran bündig in die Platinenunterseite eingesetzt und verlötet. Vor dem eigentlichen Verlöten des Transceivermoduls wird die

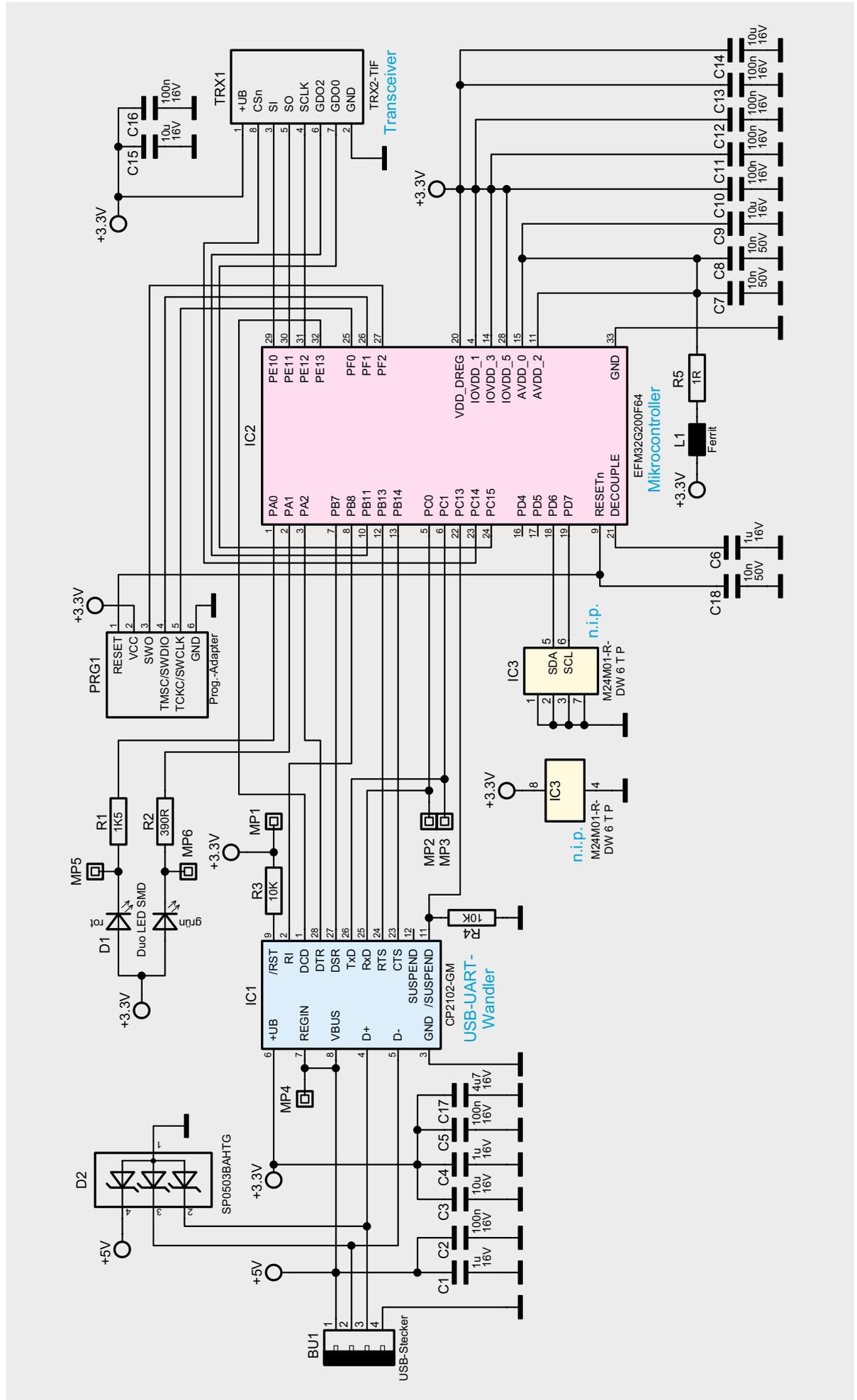


Bild 1: Das Schaltbild des HmIP-RFUSB

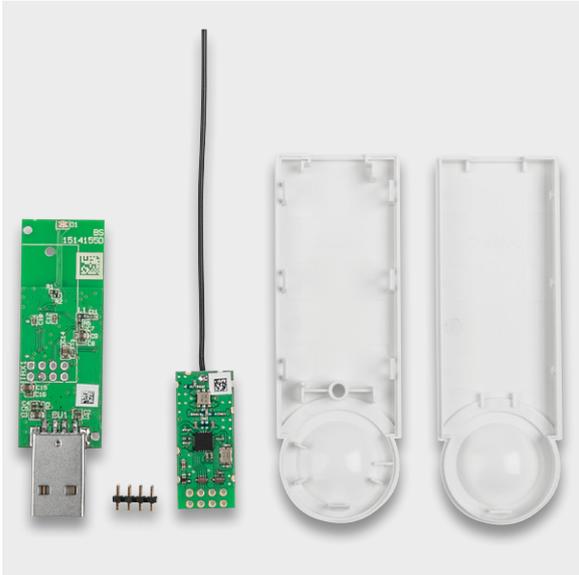


Bild 2: Der vollständige Lieferumfang des Bausatzes

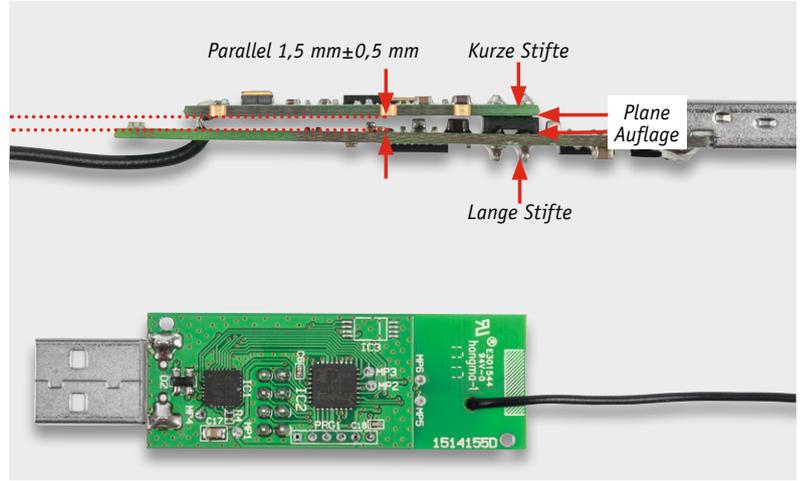


Bild 4: So erfolgt das Einsetzen und Verlöten des Transceivermoduls. Unten ist die durch die Platine geführte Antenne zu sehen.

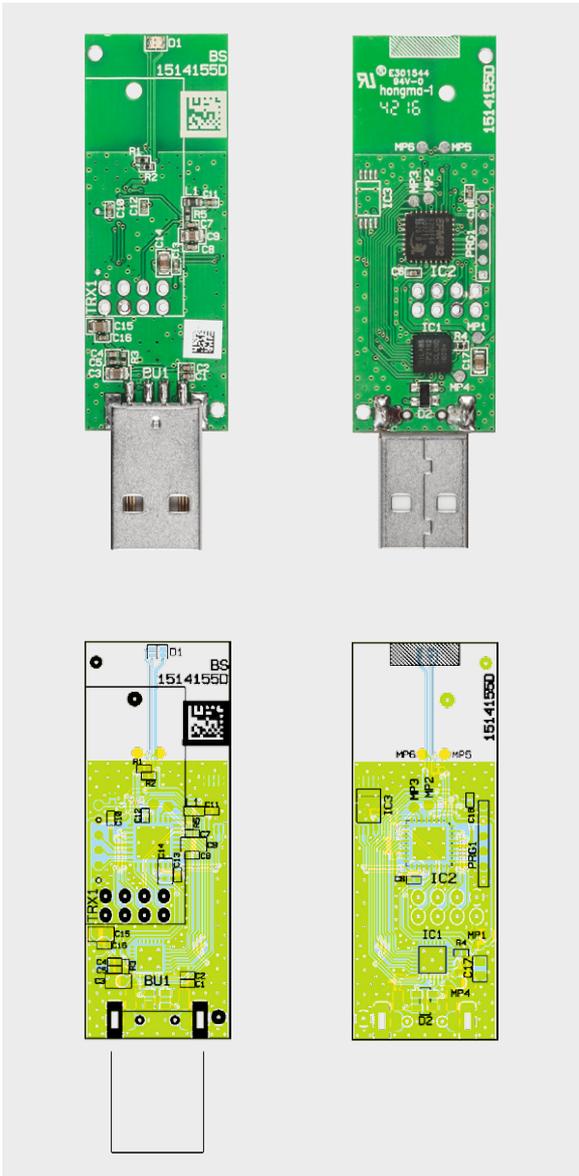


Bild 3: Platinenfotos sowie die zugehörigen Bestückungspläne der Platine

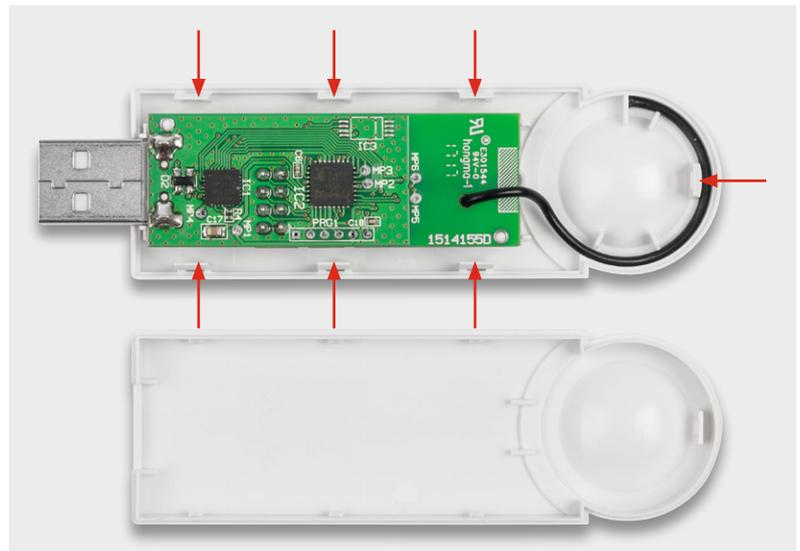


Bild 5: Die Antenne ist wie hier gezeigt in den Antennenhalter einzulegen. Hier sind auch die Gehäuseersten für das spätere Zusammsetzen der Gehäuseeile markiert.



Bild 6: Das Geräteetikett wird wie hier zu sehen, auf das Gehäuseunterteil geklebt.



Bild 7: Nach dem Zusammensetzen beider Gehäuseeile und Verrasten der Gehäuseersten (siehe Bild 5) ist das Gerät betriebsfertig.



Antenne durch das Loch der Leiterplatte auf die Platineoberseite geführt. Nun wird das Transceivermodul TRX1 bündig auf die Stiftleistenstifte aufgesetzt und verlötet (Bild 4).

Nun wird die Antenne des Transceivermoduls, wie in Bild 5 zu sehen, auf dem Halter positioniert. Dabei ist besonders darauf zu achten, dass die Antenne nicht beschädigt wird. Dem folgt das Vorbereiten des Gehäuses für den Einbau der fertig montierten Platine. Dafür ist zu allererst das Aufkleben des beiliegenden Geräteetiketts an die geeignete Stelle des Gehäuseunterteils, wie in Bild 6 zu sehen, vorzunehmen.

Sitzt die Platine in der vorgesehenen Position, wird die Gehäuseoberschale auf die gegenüberliegende Seite aufgesetzt und eingerastet (Bild 7). Somit ist das Gerät vollständig montiert und einer Inbetriebnahme steht nichts mehr im Wege.



#### Widerstände:

1 $\Omega$ /SMD/0402	R5
390 $\Omega$ /SMD/0402	R2
1,5 k $\Omega$ /SMD/0402	R1
10 k $\Omega$ /SMD/0402	R3, R4

#### Kondensatoren:

10 nF/50 V/SMD/0402	C7, C8, C18
100 nF/16 V/SMD/0402	C2, C5, C10-C13, C16
1 $\mu$ F/16 V/SMD/0402	C1, C4, C6
4,7 $\mu$ F/16 V/SMD/0805	C17
10 $\mu$ F/16 V/SMD/0805	C3, C9, C14, C15

#### Halbleiter:

ELV141379/SMD/USB-Controller	IC1
ELV141380/SMD	IC2
Duo-LED/rot/grün/SMD	D1
SP0503BAHTG/SMD	D2

#### Sonstiges:

Chip-Ferrit, 600 $\Omega$ bei 100 MHz, 0603	L1
USB-Einbaustecker, abgewinkelt, ultraflach, SMD	BU1
Stiftleiste, 2x 4-pol., gerade	TRX1
Sender-/Empfangsmodul TRX2-TIF	TRX1
Gehäuseoberteil	
Gehäuserückteil, bedruckt	
USB-Kabel	
QR-Code-Aufkleber für HmIP Geräte, weiß	

Stückliste

## Inbetriebnahme

Für die Inbetriebnahme ist der HmIP-RFUSB Stick an einen USB-Port eines PCs anzuschließen. Sollte es nicht möglich sein, den HmIP-RFUSB Stick direkt im USB-Port eines PCs anzuschließen, kann man das beiliegende Verlängerungskabel verwenden. Ausschließlich dieses Kabel ist zu nutzen, um das Gerät abgesetzt vom PC zu platzieren. Der PC erkennt den HmIP-RFUSB Stick automatisch, somit ist keine separate Treiberinstallation notwendig.

Die für den Betrieb erforderliche Software ist Bestandteil des SDK und DDK. Bei deren Installation ist den Anweisungen am Bildschirm zu folgen.

Alles Weitere ist in den zu den Entwicklungstools gehörenden Dokumentationen sowie im zum Bausatz gelieferten Bedienhandbuch zu finden. **ELV**



## Weitere Infos:

- [1] GPIO-Funkmodul: <https://www.elv.de>: Webcode #10141
- [2] OCCU-Github-Repository: <https://github.com/eq-3/occu>
- [3] Windows-Treiber-Download: [www.elv.de](http://www.elv.de): Webcode #10148
- [4] Homematic Manager: <https://github.com/hobbyquaker/homematic-manager>
- [5] YAHM: <https://homematic-forum.de/forum/viewforum.php?f=67>
- [6] RaspberryMatic: <https://github.com/jens-maus/RaspberryMatic>
- [7] Homematic Funk-LAN-Gateway: <https://www.elv.de>: Webcode #10142
- [8] Docker-Container-Website: <https://github.com/litti/dccu2>
- [9] Homematic Forum zu Docker-Container: <https://homematic-forum.de/forum/viewforum.php?f=54>
- [10] Docker-Community-Edition (CE): <https://www.docker.com/community-edition#/download>
- [11] Docker-Dokumentation: <https://docs.docker.com/>
- [12] Dokumentation Synology NAS: [https://www.synology.com/de-de/knowledgebase/DSM/help/Docker/docker\\_container](https://www.synology.com/de-de/knowledgebase/DSM/help/Docker/docker_container)
- [13] Dokumentation QNAP-NAS: <https://www.qnap.com/de-de/how-to/tutorial/article/verwendung-der-container-station>
- [14] Projektseite Docker-CCU2: <https://github.com/angelnu/docker-ccu2>