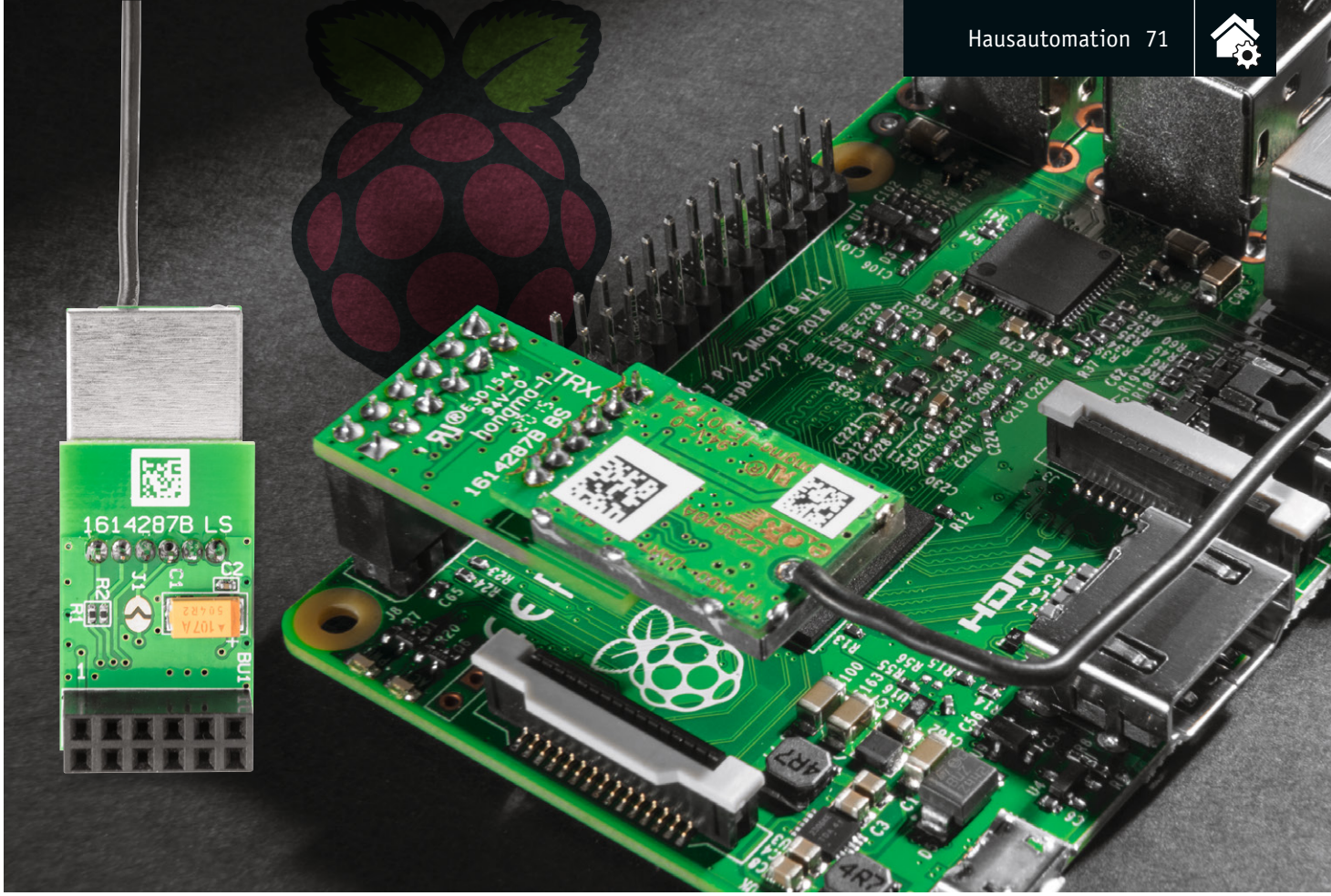


Machen den Raspberry Pi zur HomeMatic-Zentrale – HomeMatic®-Funkmodul und HM-OCCU-SDK

Infos zum Bausatz
im ELV-Web-Shop
#1385

Der beliebte Mini-Computer Raspberry Pi bildet, ausgestattet mit einem zusätzlichen Funkmodul und dem HomeMatic-Open-Central-Control-Unit-Software-Development-Kit, kurz HM-OCCU-SDK, eine alternative Plattform gegenüber der HomeMatic-Zentrale CCU2. So kann man die von der CCU2 gewohnten HomeMatic-Funktionen mit dem Raspberry Pi nutzen bzw. neue Softwarelösungen erstellen.

Raspberry Pi statt CCU?

Warum nicht? Die CCU2 basiert genau wie die vielen ARM-Minirechner ebenfalls auf einer ARM-Plattform, auf der eine Linux-Distribution läuft. Hier wie dort erfolgen Zugriff, Steuerung und Kommunikation Web-Browser-basiert, und die Software-Kommunikation wird über offengelegte Schnittstellen abgewickelt. Open-Source-Plattformen wie der Raspberry Pi sind, auch mit ihren vielen Hardware-Schnittstellen, natürlich vielseitiger einsetzbar als die eher proprietäre CCU2-Hardware, und man besitzt die Option, bei steigenden Anforderungen relativ einfach eine leistungsfähigere Hardware einzusetzen.

Da auch die Software der CCU2 seit April 2015 für andere alternative Hardware-Plattformen geöffnet wurde und somit Entwicklern und Anwendern frei als HM-OCCU-SDK zur Verfügung steht, erhält der ambitionierte Anwender hier den kompletten Werkzeugkasten für die Realisierung einer individuellen HomeMatic-Zentrale.

Technische Daten

Geräte-Kurzbezeichnung:	HM-MOD-RPI-PCB
Versorgungsspannung:	1,8–3,6 VDC
Stromaufnahme:	50 mA max.
Funkfrequenz:	868,3 MHz/869,525 MHz
Empfängerkategorie:	SRD category 2
Typ. Funk-Freifeldreichweite:	> 100 m
Duty Cycle:	< 1 % pro h/< 10 % pro h
Umgebungstemperatur:	-10 bis +55 °C
Abmessungen (B x H x T):	19 x 41 x 14 mm
Gewicht:	6 g

Das Funkmodul

Das hier vorgestellte Funkmodul für den Raspberry Pi stellt die Verbindung zwischen der CCU2-Software auf dem Raspberry Pi und HomeMatic-Funkkomponenten her. Es wird nach dem Zusammenbau einfach auf die GPIO-Steckerleiste des Raspberry Pi aufgesetzt – schon ist die Zentralen-Hardware fertig. Das Modul und das Softwarepaket sind kompatibel zum Raspberry Pi 2B.

Die Platine bildet die Schnittstelle zwischen der UART-Schnittstelle des Raspberry Pi und dem Standard-Sende-/Empfangsmodul des HomeMatic-Systems.

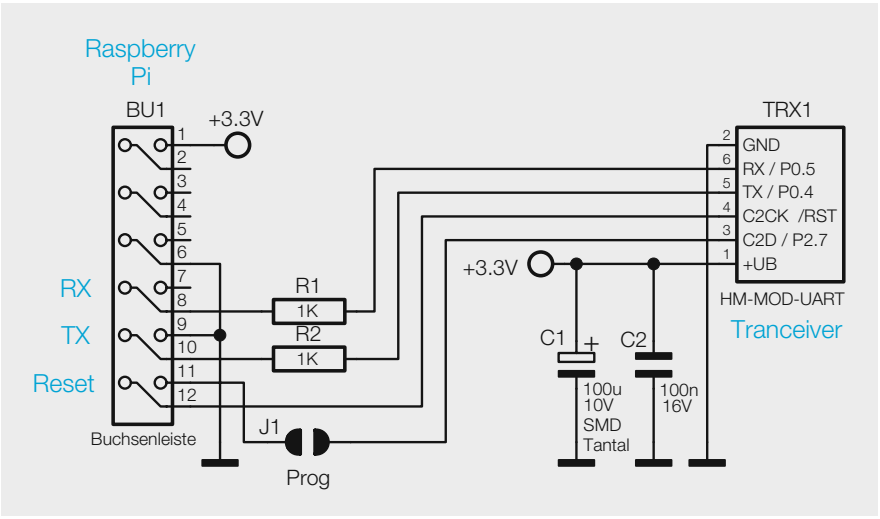


Bild 1: Das Schaltbild des Funkmoduls

Montagevideo

#1381

QR-Code scannen oder Webcode im Web-Shop eingeben

Die Schaltung, in Bild 1 zu sehen, beinhaltet neben dem Funkmodul TRX1 lediglich zwei Kondensatoren C1 und C2 zur Spannungsstabilisierung und zwei Widerstände in den RX- und TX-Leitungen zum Schutz des Funkmoduls. Über die Signalleitung C2CK /RST lässt sich ein Reset des Moduls auslösen. Die Jumperfläche J1 kann bei Bedarf den Programmierpin des TRX-Moduls zugänglich machen. Zur Vermeidung versehentlicher Programmiervorgänge ist der Pin C2D des TRX-Moduls durch den Jumper getrennt.

Über die Buchsenleiste BU1 wird das Modul auf den GPIO des Raspberry Pi gesteckt.

Für die Ansteuerung des Funkmoduls steht ein angepasstes Softwarepaket für den Raspberry Pi als Download zur Verfügung.

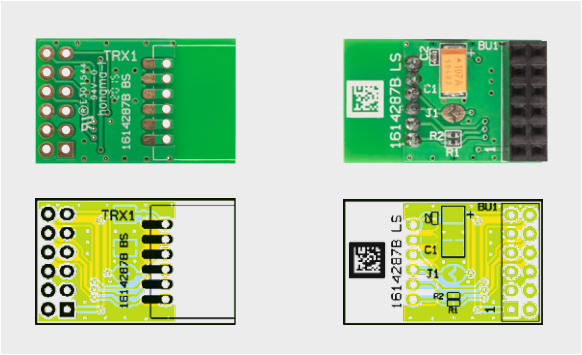


Bild 2: Platinenfoto und Bestückungsplan des Funkmoduls

Nachbau

Beim Nachbau des Funkmoduls sind lediglich die Stiftleiste vom TRX1 sowie das Funkmodul TRX1 selbst und die Buchsenleiste BU1 entsprechend Platinenfoto und Bestückungsvorgabe in Bild 2 zu bestücken. Die weiteren Komponenten sind in SMD-Technik ausgeführt und bereits vorbestückt.

Zuerst wird die Stiftleiste in das TRX1-Modul eingelötet, siehe Bild 3, dabei ist darauf zu achten, dass die Stiftleiste plan aufliegt.

Die Buchsenleiste BU1 wird von unten in die Platine eingesetzt und auf der Oberseite verlötet, diese muss ebenso plan auf der Platine aufliegen (Bild 4).

Anschließend wird das so vorbereitete Funkmodul in die Basisplatine in der markierten Fläche von oben eingesetzt und auf der Unterseite verlötet. Das Funkmodul sollte dabei plan auf der Platine aufliegen (Bild 5).

Nach der Kontrolle auf Bestückungs- und Lötfehler kann die Inbetriebnahme erfolgen. Bild 6 zeigt das fertig bestückte Modul.

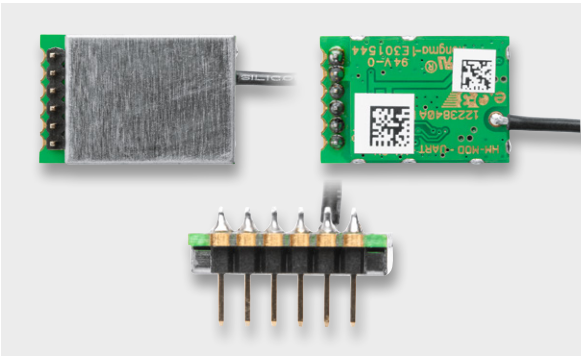


Bild 3: Im ersten Schritt erfolgt das Einlöten der Stiftleiste in die TRX-Platine. Hier ist auf planes Einsetzen der Stiftleiste zu achten.

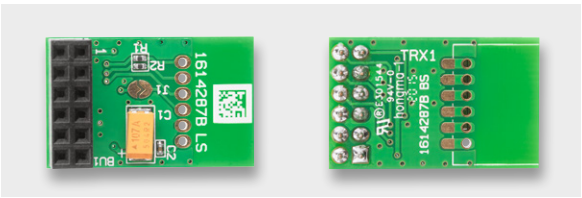


Bild 4: Im zweiten Schritt ist die Buchsenleiste in die Platine einzulöten.

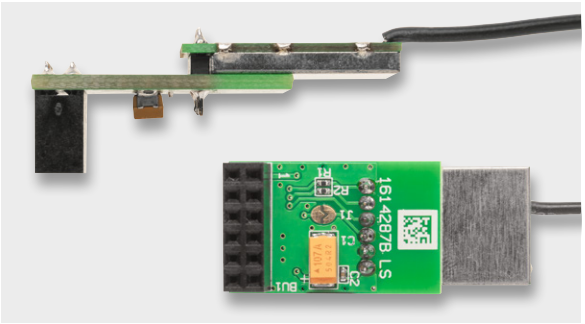


Bild 5: Das TRX-Modul wird ebenfalls mit der Stiftleiste plan in die Platine eingesetzt und verlötet.

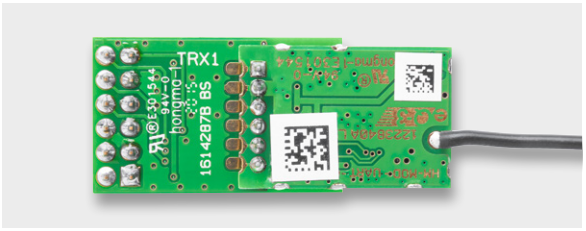


Bild 6: Das komplett bestückte Funkmodul

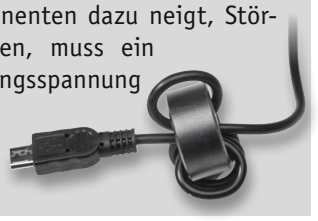
Installation auf dem Raspberry Pi

Das Modul darf nur auf den Raspberry Pi aufgesetzt werden, wenn dieser spannungslos ist. Es ist, wie in Bild 7 zu sehen, auf die Pins 1 bis 12 des Expansion-Headers zu stecken.

Die Antenne ist wie in Bild 7 und Bild 8 zu verlegen. Sie muss nach ca. 1,5 cm geknickt und parallel zur Platine geführt werden. Die Antenne darf keine anderen Bauteile berühren und sollt in einem Gehäuse fixiert werden. Bild 9 zeigt schließlich die in ein übliches Raspberry Pi-Gehäuse eingebaute Kombination. Der Einbau in ein Gehäuse ist aus EMV-Gründen und zur Vermeidung von Kurzschlüssen, Berührungen usw. unbedingt zu empfehlen.

Achtung – Störaussendung:

Beim Einsatz des Moduls mit einem Raspberry Pi ist im Hinblick auf die Störaussendung noch etwas zu beachten. Da der Raspberry Pi beim Anschluss von externen Komponenten dazu neigt, Störsignale über die Zuleitung auszusenden, muss ein Ferritring in die Zuleitung der Versorgungsspannung eingebracht werden. Hierzu wird die Zuleitung viermal durch den Ferritring geführt. Das Bild zeigt ein Beispiel dazu.



Die Software

Die Software für den Raspberry Pi basiert auf dem HomeMatic-Open-Central-Control-Unit-Software-Development-Kit, kurz HM-OCCU-SDK, das die eQ-3 AG Anfang April 2015 freigegeben hat. Die Software kann von der Webseite der eQ-3 AG [1] heruntergeladen werden und steht unter einer sehr liberalen Lizenz zur freien Verfügung. Die Software darf in Open-Source-Projekten wie auch in kommerziellen Produkten verwendet werden.

Zusätzlich gibt es ein GitHub-Projekt [2], wo aktuelle Änderungen aus der CCU2-Entwicklung und der Open-Source-Community zusammenfließen. Änderungen, die für den Raspberry Pi notwendig sind, finden sich hier ebenfalls. Das GitHub-Projekt ist in der Regel aktueller, da dort häufiger Änderungen veröffentlicht werden, wohingegen auf der eQ-3 Webseite nur neue Releases der HM-OCCU-SDK zur Verfügung stehen.

Das SDK besteht aus folgenden Teilen:

- Binärpakete für verschiedene CPU Architekturen
- CCU2 WebUI
- CCU2 Open Source Files und Toolchain

Die freigegebene Software bietet die Möglichkeit, die von der CCU2 gewohnten HomeMatic-Funktionen auf anderen Plattformen zu nutzen bzw. neue Softwarelösungen zu erstellen, die nur einige Module (z. B. den rfd) aus dem HM-OCCU-SDK verwenden und mit anderen Open-Source-Komponenten kombiniert werden (z. B. das Hmcon-Projekt, ebenfalls auf GitHub unter [3] zu finden).

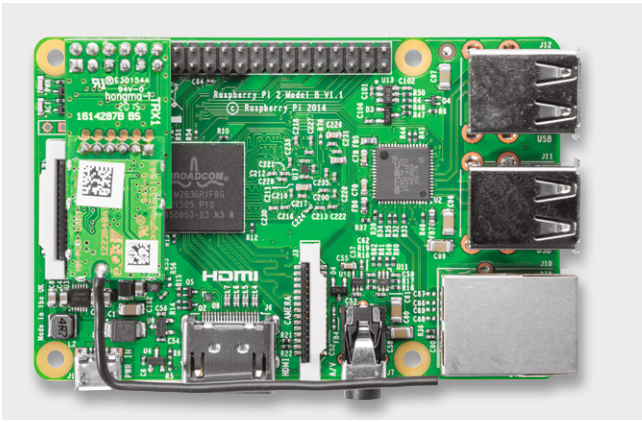


Bild 7: Das Funkmodul wird auf die GPIO-Positionen 1–12 gesteckt.

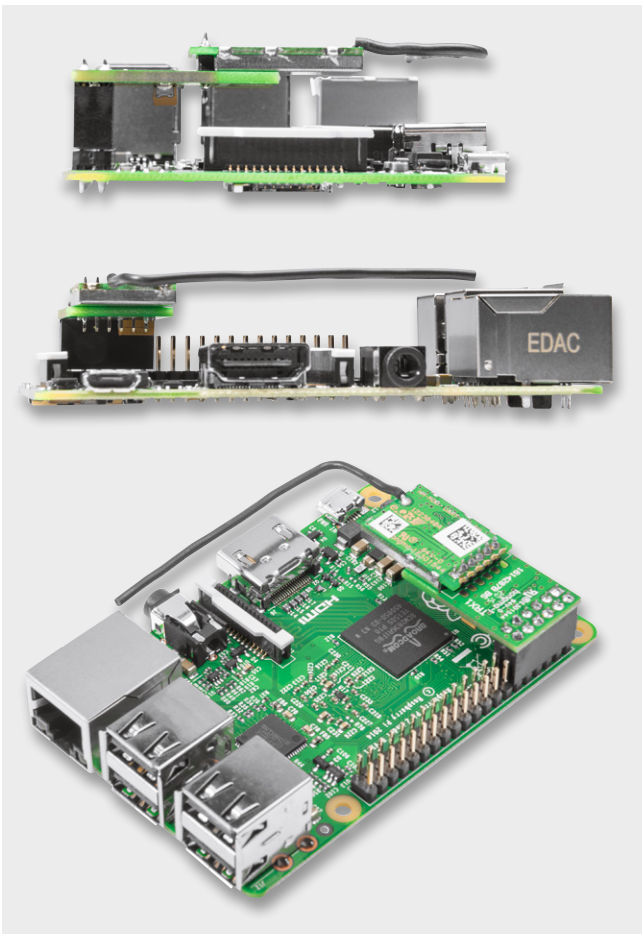


Bild 8: Verlegung der Antenne

Bild 9: Die fertige Kombination aus Raspberry Pi und Funkmodul sollte in ein passendes Gehäuse eingebaut werden.

Für den Raspberry Pi werden die Binärpakete unter „arm-gnuebihf“ benötigt.

Einordnung in bestehende Open-Source-Projekte

Abgesehen von der HM-OCCU-SDK gibt es zurzeit zwei weitere Ansätze, die CCU2-Softwarekomponenten auf einem Raspberry Pi zu betreiben:

Der erste Ansatz basiert auf einer reinen Kopie der Dateien auf eine vorbereitete SD-Karte [4]. Dieser Ansatz aus dem HomeMatic-Forum vom Foren-User „IrinaMueller” machte das erste Mal deutlich, dass die CCU2-Software ohne weitere Änderungen an den Binärpaketen auch auf anderer Hardware als der CCU2 lauffähig ist. Dieser Ansatz wird jedoch nicht mehr aktiv gepflegt, da nach jedem Firmware-Update erneute Änderungen an den Quelldateien (vor allem der WebUI) erforderlich sind. Er ist daher lediglich als „proof of concept“ zu verstehen, das sicherlich mit dazu beigetragen hat, dass die heutige HomeMatic-OCCU-SDK existiert. Dieser Ansatz sollte jedoch nicht für Produktiv-Systeme verfolgt werden.

Der zweite Ansatz wird ebenfalls von einigen Foren-Usern aus dem HomeMatic-Forum verfolgt. Er wird unter dem Namen LXCCU beschrieben [5]. Bei diesem Ansatz wird ebenfalls, wie im ersten Ansatz, die CCU2-Firmware verwendet und verändert. Die Software läuft jedoch statt auf der eigentlichen Hardware in einem LXC-Container (Erklärung unter [6]). Der Vorteil ist die Hardwareunabhängigkeit – so kann die CCU2-Firmware auf jedem beliebigen ARM-System eingesetzt werden. Der Nachteil ist jedoch, dass die Unterstützung von LXC durch den jeweils eingesetzten Kernel gegeben sein muss. Zusätzlich entsteht ein etwas höherer Verwaltungsaufwand durch die weitere Abstraktionsschicht (den Container). Da auch hier die Quelldateien der Firmware nach jedem Release zurzeit immer noch angepasst werden müssen, kann es nach Erscheinen einer neuen Firmware einige Zeit dauern, bis diese im Container [7] eingesetzt werden kann.

Generelles Ziel all der beschriebenen Lösungen ist übrigens keine Kosteneinsparung gegenüber der Anschaffung einer CCU2, sondern die Leistungssteigerung der HomeMatic-Zentralenlösung gegenüber der Original-CCU2.

Derzeit läuft die Bereitstellung von immer neuen ARM-embedded-Rechnerplattformen rasant, jede Neuerscheinung hat eine höhere Rechenleistung und mehr Datendurchsatz, und so kann die Performance insbesondere in größeren HomeMatic-Installationen erheblich gesteigert werden.

Die Lösung unter HM-OCCU-SDK

Ziel des hier vorgestellten Ansatzes mit der HM-OCCU-SDK ist es, eine Software zur Verfügung zu stellen, die Out-of-the-box ohne weitere Anpassungen direkt auf einem Raspberry Pi eingesetzt werden kann. Dabei kann die bestehende HomeMatic-Infrastruktur (Funk-Gateway, Wired-Gateway) genutzt werden oder die Aufsteckplatine zum Einsatz kommen.

Bild 10 zeigt die Software-Architektur der CCU2, die für dieses Projekt auf den Raspberry Pi angepasst wurde.

Die unterste Schicht bilden die sogenannten Schnittstellenprozesse. Neben dem HomeMatic-Funk (rfd), gibt es eine weitere Schnittstelle zu HomeMatic Wired und ab Herbst 2015 ebenfalls zu den neuen Homematic-IP-Funkkomponenten. Die Schnittstellenprozesse haben die Aufgabe, die unterschiedlichen Transportprotokolle (HomeMatic-Funk, HomeMatic Wired, Homematic-IP-Funk) in ein einheitliches Format umzusetzen und den höheren Schichten über eine Schnittstelle (XML-RPC) zur Verfügung zu stellen. Neben den HomeMatic-Schnittstellenprozessen gibt es den CUxD als weitere Schnittstelle. Über den CUxD können unter anderen ausgewählte FS20-Geräte an die CCU2 angebunden werden. Die Funktionen der XML-RPC-Schnittstelle wurden bereits vor Jahren veröffentlicht [8], sodass es mittlerweile viele Erweiterungen gibt, die diese Schnittstelle verwenden.

Wer den Schnittstellenprozess rfd für HomeMatic-Funk in eigene Lösungen einbinden will, findet in Tabelle 1 („Beschreibung der Startparameter für den Schnittstellenprozess rfd“) eine ausführliche Beschreibung.

Da die Schnittstellenprozesse die unterschiedlichen Transportprotokolle umsetzen, beschränken sich die Informationen, die über die XML-RPC Schnittstelle abgefragt werden können, auf technische Werte wie zum Beispiel Seriennummer, Ist- bzw. Solltemperatur beim Heizkörperthermostaten. Informationen wie Räume, Gewerke, Bezeichnungen wie „Heizung Bad“ fehlen. Diese Informationen werden in der nächst höheren Schicht, dem sogenannten Backend, verwaltet.

Auf der CCU2 ist hierfür das Programm ReGaHss zuständig. Das Programm liest beim Start die Datei /etc/config/InterfacesList.xml aus. In dieser Datei sind alle Schnittstellenprozesse aufgeführt, die über die XML-RPC-Schnittstelle zur Verfügung stehen. ReGaHss fragt beim Start bei allen Schnittstellenprozessen die Liste der bekannten Geräte ab und erhält ab diesem Zeitpunkt von den Schnittstellenprozessen Nachrichten, sogenannte Events. Wenn z. B. eine Taste auf einer Funkfernbedienung gedrückt wird, wird das dem Backend (ReGaHss) mitgeteilt. Solche Events können dann in den „Wenn-Dann-Sonst“-Programmen der WebUI ausgewertet werden.

Neben diesen Logikfunktionen stellt das Programm ReGaHss noch folgende Funktionen zur Verfügung:

a) An- und Ablernen von Geräten
b) Verwaltung von direkten Geräte-Verknüpfungen
c) Verwaltung von Metadaten wie zum Beispiele Raum- und Gerätenamen
d) Ausführen von HomeMatic-Script-Programmen

Die Funktionen sind im „Handbuch HomeMatic WebUI“ [9] genau beschrieben, einen schnellen Überblick über die wichtigsten Funktionen findet man auch im „HomeMatic Systemkurzleitfaden“ [10].

Neben den Funktionen a bis d enthält ReGaHss auch einen Webserver, der von einigen Webseiten der WebUI genutzt wird (z. B. Wenn-Dann-Sonst-Programmierung). Andere Webseiten werden durch normale CGI Scripte (tcl) zur Verfügung gestellt. Ein Großteil der Anpassungen, die für den Betrieb auf

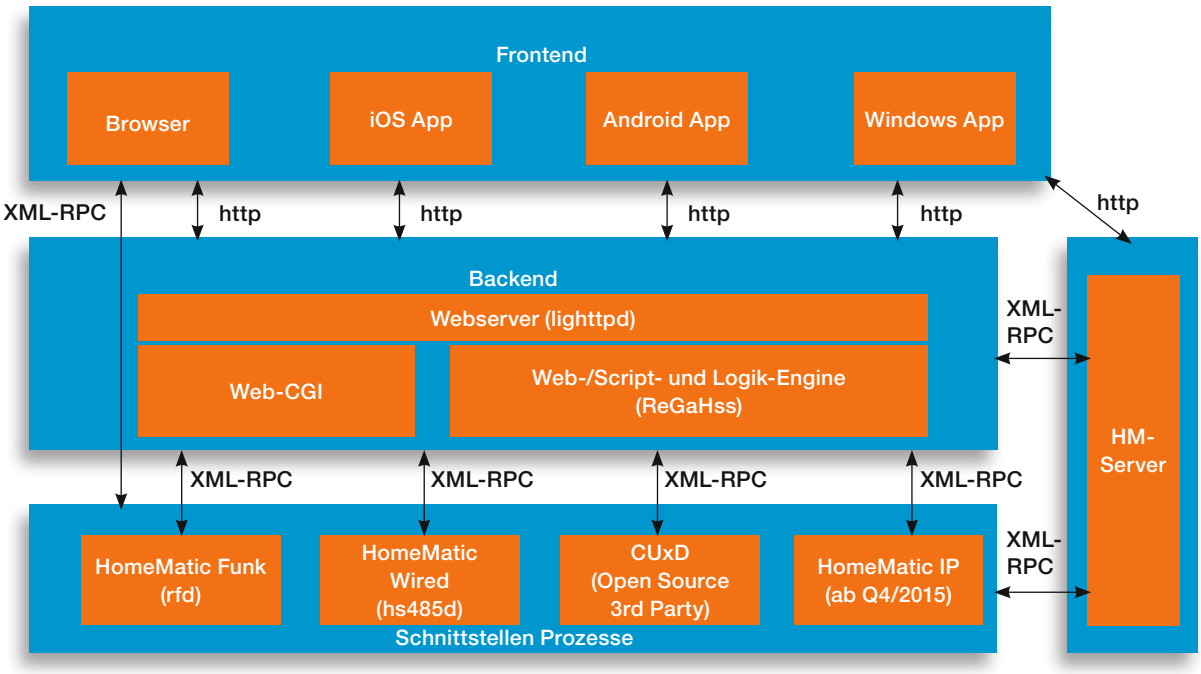


Bild 10: Die Software-Architektur der CCU2, die für dieses Projekt angepasst wurde, mit Drittanbieter-Programmen

dem Raspberry Pi notwendig waren, wurde in den tcl-CGI-Skripten vorgenommen.

Eine Besonderheit stellt der HM-Server auf der CCU2 dar. Neuere Funktionen wie das Gruppenkonzept und die Messdatenerfassung wurden nicht mehr im „alten" ReGaHss-Backend implementiert.

Der HM-Server stellt folgende Funktionen zur Verfügung:

a) virtuelle Geräte (Heizgruppen)
b) Speicherung der Messdaten in einer RRD-Datenbank (Round-Robin-Database)
c) Webseiten für die Anzeige der Messdaten sowie einige Systemfunktionen (SD-Karte)

Damit ist der HM-Server eine Mischung aus Backend und Schnittstellenprozess.

Software Installation auf einem Raspberry Pi

Die Software für den Raspberry Pi wird in zwei verschiedenen Formen zur Verfügung gestellt: zum einen als Debian-Paket (deb Files) über ein eigenes Repository. Das hat den Vorteil, dass die Software auf bereits existierenden Installationen eingesetzt werden kann. Zum anderen gibt es ein spezielles Image, das auf einer SD-Karte installiert werden kann.

Für die Installation existiert eine separate Installationsanleitung, die unter [11] zu finden ist.

Nach der Installation lässt sich über einen Browser die von HomeMatic bekannte Weboberfläche aufrufen. Dort können, wie aus HomeMatic gewohnt, Geräte angelernt, konfiguriert und bedient werden. ELV

Tabelle 1

Beschreibung der Startparameter für den Schnittstellenprozess rfd

Der rfd bietet Unterstützung für folgende Startparameter:

-d	*Startet den rfd im Hintergrund (daemonize)*
-c	*Log-Ausgabe auf der Console*
-f [Datei]	*Gibt den Speicherort der Konfigurationsdatei (rfd.conf) an*
-l [Loglevel]	*Gibt das Loglevel an (0 -> Loggen, 6 -> nur Fatale Fehler)*

Die Logikschicht ReGaHss bietet Unterstützung für folgende Parameter:

-f [Datei]	*Gibt den Speicherort der Konfigurationsdatei (rega.conf) an*
-l [Loglevel]	*Gibt das Loglevel an (0 -> kein Logging bis 5 -> alles loggen)*

Die Konfigurationsoptionen der Konfigurationsdatei rfd.conf erklärt:

Die Datei rfd.conf besteht aus Schlüssel-Wert-Paaren. Ein Paar ist dabei in jeweils einer Zeile geschrieben. Der Schlüssel steht dabei am Zeilenanfang. Dann folgt ein Leerzeichen, das Gleichheitszeichen und erneut ein Leerzeichen. Am Zeilenende steht der Wert.
Zeilen können mit einer Raute am Zeilenanfang auskommentiert werden. Diese werden dann nicht verwendet.

→

Allgemeine Konfigurationsoptionen:

Listen Port	*TCP-Port für XmlRpc-Verbindungen*
Listen IP	*IPv4-Adresse des Hosts, auf die der rfd horcht*
Log Destination	*Ort der Log-Ausgabe (gültige Werte: None, Syslog oder File)*
Log Filename	*Dateipfad der Logdatei, wenn Log Destination = File*
Log Level	*Umfang der Log-Ausgabe (gültige Werte: 0–6; 0=ALL, 1=DEBUG, 2=WARNING, 3=INFO, 4=NOTICE, 5=WARNING, 6=ERROR)*
Persist Keys	*Wenn 1, werden die AES-Schlüssel in einer Datei gespeichert (dringend empfohlen; gültige Werte: 0 oder 1)*
Key File	*Pfad zur Key-Datei wenn Persist Keys = 1*
Address File	*Pfad zu der Datei, die die eindeutige Funkadresse enthält*
Device Description Dir	*Pfad zum Verzeichnis, welches die XML-Gerätebeschreibung enthält*
Device Files Dir	*Pfad zum Verzeichnis, in dem der rfd die Geräteprofile speichert*
Firmware Dir	*Pfad zum Verzeichnis, in dem der rfd nach Gerätefirmware-Updates sucht*
User Firmware Dir	*Pfad zum Verzeichnis, in dem der rfd nach vom Benutzer eingespielten Gerätefirmwares sucht*
Remove Unreachable Clients	*Gibt an, ob unerreichbare Clients (zum Beispiel nicht mehr erreichbare Apps, da sie sich außerhalb des lokalen WLANs befinden) nach 10 erfolglosen Versuchen aus der Liste der aktiven Clients gelöscht werden (gültige Werte: true oder false; es wird dringend geraten, nicht mehr erreichbare Clients durch true aus der Liste entfernen zu lassen (Standardoption)*
XmlRpcHandlersFile	*Pfad zur XML-RPC-Handlers-Datei*

Konfigurationsoptionen für den eingesetzten Funk-Hardwareadapter:

Die Konfigurationsdatei für den rfd muss mindestens einen Interface-Eintrag haben. Ein Interface-Eintrag wird immer mit einer Zeile
[Interface interfaceNr] eingeleitet, wobei interfaceNr durch eine eindeutige Nummer ersetzt wird (zum Beispiel [Interface 0]).
Abhängig von der Art des Interfaces folgen dann unterschiedliche Schlüssel-Wert-Paare.

Interface-Konfigurationsoptionen für CCU2/HM-MOD-UART:

Type	*Interface-Typ für CCU2/HM-MOD-UART (immer: CCU2)*
ComPortFile	*Datei für seriellen Port/COM-Port*
AccessFile	*Pfad zur Gerätedatei (dev-File), die Zugriff auf den Reset-Pin gibt (kann auf /dev/null oder eine leere Datei zeigen, wenn nicht gültig)*
ResetFile	*Pfad zur Gerätedatei (dev-File) die den Reset des Coprozessors/Moduls auslöst (kann auf /dev/null oder eine leere Datei zeigen, wenn nicht gültig)*
Serial Number	*Überschreibt die Seriennummer des Coprozessors (optional)*

Interface-Konfigurationsoptionen für HM-Cfg-USB und HM-Cfg-USB-2:

Type	*Interface-Typ (immer: USB Interface)*
Description	*Beschreibung (optional)*
Serial Number	*Seriennummer des Adapters, zum Beispiel ABC1234567*

Interface-Konfigurationsoptionen für HM-LGW-O-TW-W-EU:

Type	*Interface-Typ (immer: HMLGW2)*
Description	*Beschreibung (optional)*
Serial Number	*Seriennummer des Adapters, zum Beispiel ABC1234567*
Encryption Key	*geheimer Schlüssel (Aufkleber auf Adapter)*

Beispiel:

```
# TCP Port for XmlRpc connections
Listen Port = 2001

# local IP to bind to (optional)
# Listen IP = 172.25.50.21

# Log Level: 1=DEBUG, 2=WARNING, 3=INFO, 4=NOTICE, 5=WARNING, 6=ERROR
Log Level = 1

# If set to 1 the AES keys are stored in a file. Highly recommended.
Persist Keys = 1

Device Description Dir = /opt/rfd/devicetypes
Device Files Dir = /etc/rfd/devices
```

Tabelle 1

→

Tabelle 1

```
Key File = /etc/rfd/keys
Address File = /etc/rfd/ids
Log Destination = Syslog
#Log Filename = /var/log/rfd

Firmware Dir = /firmware
User Firmware Dir = /rfd/user-firmware

#Interface definition for CCU 2 RF Coprocessor/ HM-MOD-UART
[Interface 0]
Type = CCU2
Description = CCU2-Coprocessor
ComPortFile = /dev/ttyAPP0
AccessFile = /dev/null
ResetFile = /dev/ccu2-ic200
#Serial Number = ABC123457

#Interface definition for HM-LGW-O-TW-W-EU Lan Gateway
#[Interface 1]
#Type = HMLGW2
#Description = HM-LGW-O-TW-W-EU
#Serial Number = ABC1231456
#Encryption Key = secretkey

#Interface definition for HM-Cfg-USB and HM-Cfg-USB-2
#[Interface 2]
#Type = USB Interface
#Description = USB Interface
#Serial Number = ABC1234567
```

Stückliste

1 kΩ/SMD/0402	R1, R2
100 nF/16 V/SMD/0402	C2
100 µF/10 V	C1
Buchsenleiste, 2x 6-pol., gerade	BU1
Modul HM-MOD-UART-AW-SH-2 eQ-3, komplett	TRX1

Weitere Infos:

[1] HM-OCCU-SDK: http://www.eq-3.de/Downloads/Software/Software_Development_Kit/HM-OCCU-SDK-1.0.0.tgz

[2] www.github.com: https://github.com/eq-3/occu

[3] GitHub Hmcon: https://github.com/hobbyquaker/hmcon

[4] HomeMatic-Forum: http://homematic-forum.de/forum/viewtopic.php?f=26&t=13303&start=30#p101875

[5] HomeMatic-Forum: http://homematic-forum.de/forum/viewtopic.php?f=26&t=18359

[6] LXCCU: http://www.lxccu.com/

[7] LXC-Container: https://linuxcontainers.org/

[8] XML-RPC Dokumentation: http://www.eq-3.de/Downloads/Software/HM-CCU2-Firmware_Updates/Tutorials/HM_XmlRpc_API.pdf

[9] Handbuch HomeMatic WebUI: http://www.eq-3.de/Downloads/PDFs/Dokumentation_und_Tutorials/WebUI_Handbuch_V3.3_web.pdf

[10] HomeMatic Kurzleitfaden: http://www.eq-3.de/Downloads/PDFs/Dokumentation_und_Tutorials/Homematic_Systemkurzleitfaden_20131219_web.pdf

[11] Installationsanleitung und Konformitätserklärung: www.elv.de: Webcode #1385