

Selbstbau CUL

Aus FHEMWiki

Ein CUL ist ein Transceiver zum Empfangen und Senden von Funknachrichten. Er besteht im Wesentlichen aus einem Atmel ATmega Mikrocontroller und einem CC1101 Funkmodul sowie einer USB-Schnittstelle zur Verbindung mit dem Host Computer auf dem z. B. FHEM läuft. Als Firmware wird die quelloffene culfw (<http://culfw.de/culfw.html>) verwendet.

Mittlerweile sind diese Baugruppen einfach und kostengünstig einzeln beschaffbar. Daher ist es relativ einfach möglich eine in der Funktionalität zum CUL vergleichbare Hardware selbst zu bauen.

Inhaltsverzeichnis

- 1 Verwendete Hardware
- 2 Die unterschiedlichen Ausführungen des Funkmoduls
- 3 Bezugsquellen
- 4 Schaltplan
- 5 Aufbau
- 6 Software
- 7 Inbetriebnahme
- 8 Hinweise zum Betrieb mit FHEM
- 9 Bekannte Probleme
- 10 Weblinks

Verwendete Hardware

Die Arduinos (<http://arduino.cc>) sind weit verbreitete Boards mit dem ATmega328p. Für den Selbstbau CUL ist insbesondere der Arduino nano (<http://arduino.cc/en/Main/ArduinoBoardNano>) gut geeignet. Er ist relativ klein, hat eine USB-Schnittstelle und kann eine 3,3V Spannungsversorgung für den CC1101 zur Verfügung stellen.

Module mit dem CC1101 Funkchip gibt es in unterschiedlichen Ausführungen. Wichtig ist die Unterscheidung nach der verwendeten Funkfrequenz, für den Einsatz in Europa sind das 433MHz und 868MHz. Zwar kann der Chip grundsätzlich auf beide Frequenzen eingestellt werden. Für eine optimale Sende- und Empfangsleistung muss aber der Antennenkreis auf die verwendete Frequenz abgestimmt werden. Damit ist nicht nur die Antenne selbst gemeint, auch die Hardware auf dem Modul ist unterschiedlich.

Zwar kann die culfw temporär auf eine andere Frequenz umschalten, aber das führt oft nur zu unbefriedigenden Ergebnissen. Es sollte daher das Modul verwendet werden, das zu dem Funkprotokoll passt welches hauptsächlich zum Einsatz kommen soll.

Die unterschiedlichen Ausführungen des Funkmoduls

Die Module gibt es von unterschiedlichen Herstellern die auch unterschiedliche Anschlüsse verwenden. Daher muss die genaue Anschlussbelegung ermittelt werden und kann nicht einfach blind aus einem Beispiel übernommen werden.

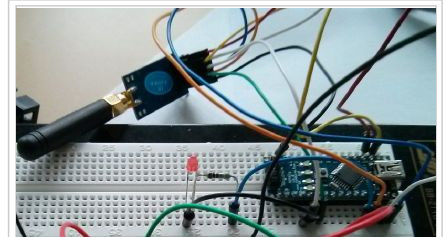
Einige Hersteller bestücken die Module statt mit dem CC1101 (<http://www.ti.com/product/CC1101/description>) mit dem CC110L (<http://www.ti.com/product/CC110L/description>), einer Low Cost Version des CC1101 mit eingeschränktem Funktionsumfang. Es ist nicht garantiert, dass alle Funktionen der culfw auch nicht dem CC110L fehlerfrei arbeiten.

Die Module für die unterschiedlichen Frequenzbereiche werden von den Anbietern leider nicht immer genau spezifiziert. Man kann die Auslegung eines Moduls anhand der Bestückung des Antennenfrontends im Vergleich zur Referenzschaltung von TI selber prüfen.

Ausgehend vom CC1101-Chip her fängt die 868Mhz Schaltung mit zwei Induktivitäten Richtung Antenne an. Optisch also zwei gleiche Bauteile. Die 433 MHz Schaltung hat zwei unterschiedliche Bauteile, eine Induktivität und einen Kondensator. Die 868 MHz Version hat dagegen zwei Induktivitäten in Reihe und dazwischen einen Kondensator auf Masse. Die 433 MHz Version hat zuerst einen Kondensator und dann eine Induktivität in Reihe sowie dazwischen noch mal einen Kondensator auf Masse.

Kondensatoren sind bräunlich, Induktivitäten weiß bzw. grün.

Selbstbau CUL



Selbstbau CUL auf Breadboard

Allgemein

Protokoll	diverse
Typ	Transceiver
Kategorie	CUL

Technische Details

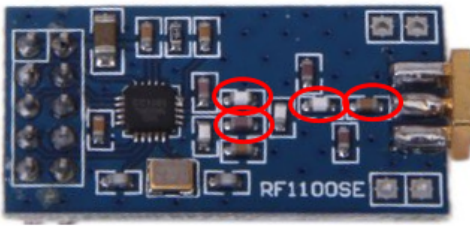
Kommunikation	Funk 433MHz oder 868MHz
Kanäle	N/A
Betriebsspannung	5V
Leistungsaufnahme	
Versorgung	USB
Abmessungen	

Sonstiges

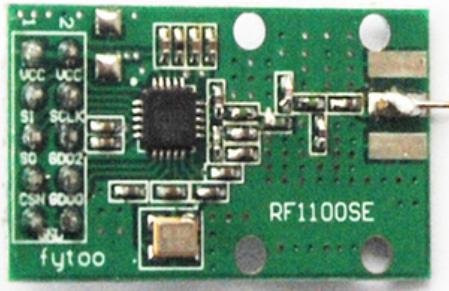
Modulname	CUL
Hersteller	du selbst

Auf diesem Bild sieht man 433Mhz Module im Vergleich zur Referenzschaltung:

RFinChina "RF1100SE" 433Mhz

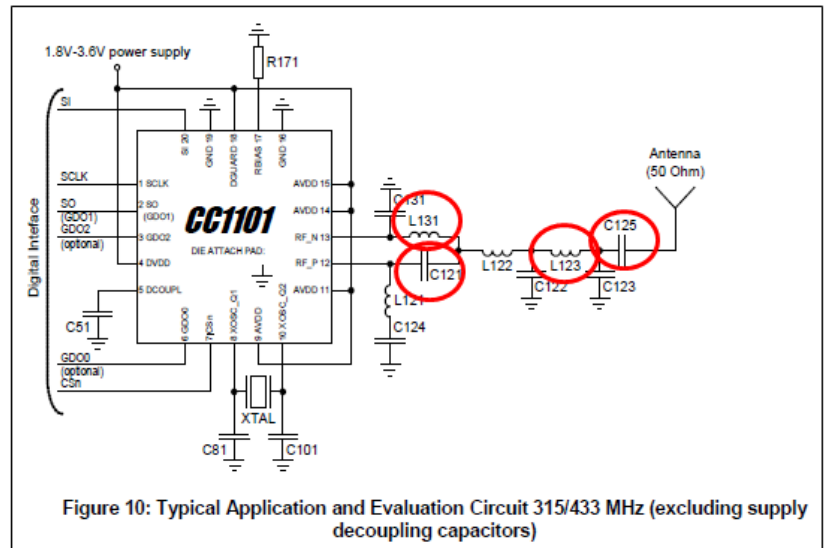


FYTOO "RF1100SE" 433Mhz (identische Bestückung)



Wird fälschlicherweise
auch als 868Mhz Modul
verkauft

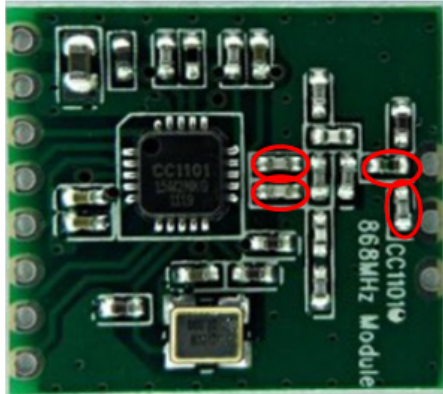
Referenzschaltung für 433Mhz von TI



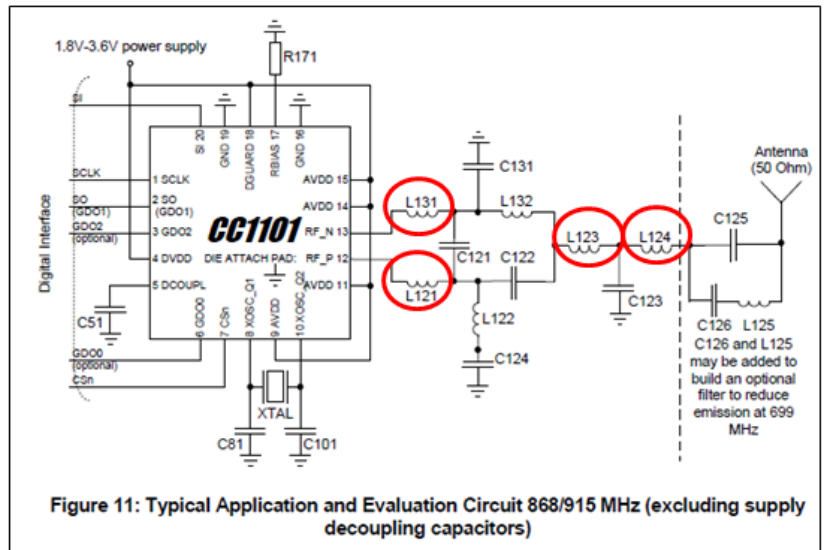
Dies sind KEINE 868Mhz Module!

Dieses Bild zeigt dagegen ein echtes 868MHz Modul:

Noname 2mm Pin Modul HF optimiert für 868 Mhz



Referenzschaltung für 868Mhz von TI



Das ist ein echtes 868Mhz Modul!

In diesem Forenbeitrag (<http://forum.fhem.de/index.php/topic,24651.msg387326.html#msg387326>) ist der Schaltplan mit Pinbelegung zum Anschließen des CC1101 mit 868Mhz verlinkt.

Bezugsquellen

Die original Arduinos sind relativ teuer. Es gibt aber billige Kopien, entweder bei den bekannten Auktionshäusern und Handelsplattformen oder direkt in China.

Dort ist ein nano für unter 5€ beschaffbar. Es sollte aber darauf geachtet werden, dass als USB-seriell Wandler auf dem nano ein FTDI FT232RL Chip oder ein anderer Chip mit eindeutiger ID verwendet wird. Nur dann sind mehrere CULs gleichzeitig ohne Probleme in FHEM nutzbar.

Auch die CC1101 Module werden dort in unterschiedlichen Ausführungen angeboten. Die 433MHz Versionen sind i.A. leichter erhältlich als die 868MHz Versionen. Wenn keine guten Lötkenntnisse vorhanden sind sollte darauf geachtet werden, ein Modul mit als Stiftleiste im 2,54mm Raster herausgeführten Anschlüssen zu verwenden.

In diesem Blog (<http://blog.gummibaer-tech.de/cul-stick-868433-im-selbstbau/>) sind konkrete Bezugsquellen genannt.

Schaltplan

Das Funkmodul ist hier nur über seine acht Anschlüsse dargestellt. Hilfreich ist der Anschluss einer LED mit passendem Vorwiderstand. Diese hilft bei der Inbetriebnahme und als grobe Funktionskontrolle.

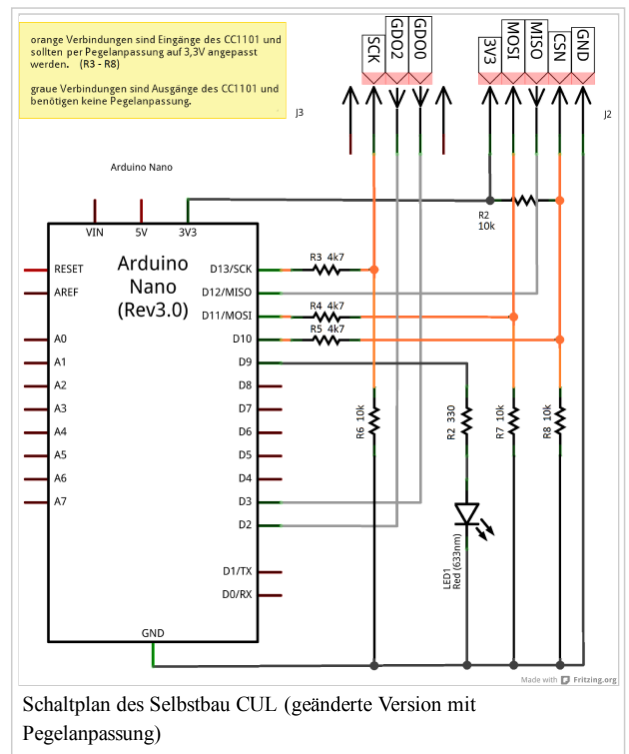
Achtung! Der CC1101 darf nur mit maximal 3.6V betrieben werden. Eine passende Versorgungsspannung stellt der Arduino nano bereit. Aber auch die Signalleitungen sind laut Datenblatt nicht 5V tolerant. Diese haben beim nano aber 5V Pegel. In der Praxis funktionieren die Module bisher trotzdem. Langzeiterfahrungen gibt es aber nicht und es ist nicht ausgeschlossen, dass die Module dadurch beschädigt werden.

Wer sicher gehen will, verwendet in den Signalleitungen Pegelanpassungen 5V -> 3,3V. Das ist mit fertigen Bausteinen (Levelshifter (<http://www.aliexpress.com/wholesale?catId=0&SearchText=i2c+level+converter>)) möglich oder im einfachsten Fall mit Widerständen (http://lowpowerlab.com/wp-content/uploads/2012/12/rfm12B-arduino-moteino-atmega328_5V_connections.png) als Spannungsteiler. Zu der Dimensionierung der Spannungsteiler siehe

? Eine mögliche Vorgehensweise, mehrere FTDI mit gleicher ID eventuell doch benutzen zu können, ist in diesem Forenbeitrag (<http://forum.fhem.de/index.php/topic,44926.msg446809.html#msg446809>) beschrieben.

auch diesen Hinweis im Forum (<http://forum.fhem.de/index.php/topic,52865.0.html>). Für eine bessere Signalübertragung sollten besser 470 Ohm/1000 Ohm genommen werden, statt der 4.7k/10k Variante in der Darstellung rechts.

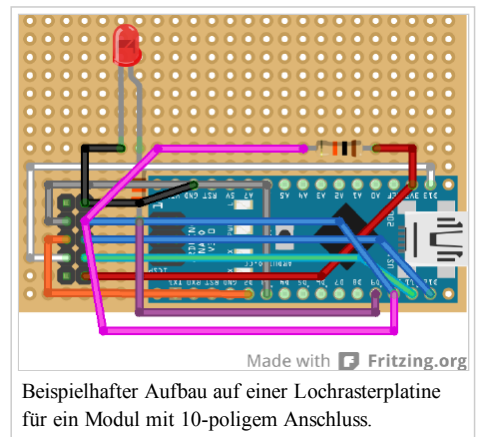
Um Empfangsproblemen vorzubeugen, sollte ein Pullup-Widerstand (R2, 10k) am CSN Pin des Moduls verwendet werden.



Aufbau

Am einfachsten ist der Aufbau auf einem Steckbrett (Breadboard). Damit lässt es sich schnell in Betrieb nehmen und mögliche Fehler können einfach korrigiert werden. Für einen permanenten Aufbau kann z. B. eine Lochraster oder Streifenrasterplatine verwendet werden.

Wer nicht löten will kann auch ein Breakoutboard mit Schraubklemmen für den Arduino nano verwenden, z. B. ein Angebot bei eBay (http://www.ebay.de/itm/121239860594?_trksid=p2059210.m2749.l2649&ssPageName=STRK%3AMEBIDX%3AIT).



Software

Um die culfw zu kompilieren, müssen ein AVR C-Cross-Compiler sowie dazugehörige Tools installiert sein. Bei einem Debian-basierten Linux geht das mit

```
sudo apt-get install make gcc-avr avrdude avr-libc
```

Anschließend die neueste Version der culfw als ZIP-Datei (<https://sourceforge.net/p/culfw/code/HEAD/tarball>) herunterladen und auspacken.

Dann in das Verzeichnis culfw/Devices/nanoCUL wechseln. In dieser Auslieferung sind alle von der culfw unterstützten Protokolle mit Ausnahme vom Wireless M-Bus aktiviert. Bei Bedarf können einzelne Protokolle durch Änderungen der entsprechenden defines in der Datei board.h aus- oder eingeschaltet werden.

Wenn ein 868MHz Modul verbaut ist, sollte in der Datei board.h die Zeile

```
#define HAS_CC1100_433
```

auskommentiert werden.

Wird ein Arduino mit einer Taktfrequenz von nur 8MHz, wie ein Pro Mini 3,3V (<http://arduino.cc/en/Main/ArduinoBoardProMini>), verwendet, so muss die Zeile

Es gibt mittlerweile mehrere alternative Versionen der culfw. Diese implementieren zusätzliche, noch experimentelle Features, sind dafür aber möglicherweise instabiler als die offizielle culfw. Siehe dazu auch

- erweiterte Intertechno und Tempsensoren Unterstützung (<http://forum.fhem.de/index.php/topic,35064.0.html>) und
- Bessere Unterstützung von Homematic (<http://forum.fhem.de/index.php/topic,24436.0.html>)

```
#define HAS_16MHZ_CLOCK
```

in der Datei board.h auskommentiert werden.

Mittels

```
make
```

wird dann die Übersetzung gestartet, die ohne Fehler durchlaufen sollte. Dabei wird die Datei nanoCUL.hex erzeugt.

Diese muss dann anschließend noch auf den Arduino geflasht werden.

Wenn der Arduino das einzige per USB angeschlossene Gerät mit serieller Schnittstelle ist, geht das ohne weitere Änderungen mit

```
make program
```

Hinweis: Wenn der Arduino bereits in FHEM konfiguriert ist, muss vor einem Flashen erst FHEM beendet werden, damit die Schnittstelle freigegeben wird. Andernfalls wird avrdude einen Fehler melden.

Inbetriebnahme

Nach dem Flashen sollte die LED einmal pro Sekunde blinken. Blinkt sie schneller oder langsamer, ist die Frequenz des Controllers nicht richtig eingestellt. Ursache können z.B. falsche Fuses sein, so dass der Controller den internen RC-Oszillator statt des externen 16MHz Quarz nutzt. Die culfw ist für eine Verarbeitungsgeschwindigkeit von 8MHz ausgelegt, daher wird beim nanoCUL die externe Frequenz von 16MHz direkt beim Start heruntergeteilt.

Bevor der Selbstbau CUL mit FHEM verwendet wird, kann man noch eine Funktionskontrolle mit Hilfe eines Terminalprogramms wie screen, picocom, cutecom oder minicom vornehmen. Dabei müssen die Kommunikationsparameter 38400/8N1 sein.

Nach der Eingabe von **V** und Bestätigung mit **Return** sollte die Versionsinformation ausgegeben werden. Nach **?** werden die unterstützten Protokolle ausgegeben, siehe die Dokumentation der culfw (<http://culfw.de/commandref.html>) für Details.

i Kommandos an die culfw müssen jeweils mit einem Zeilenumbruch bestätigt werden damit sie verarbeitet werden.

Um zu prüfen, ob der Empfang prinzipiell funktioniert, kann das Kommando X08 (<http://culfw.de/commandref.html#X>) verwendet werden. Wird auf der eingestellten Frequenz irgendwas empfangen, erfolgt eine Ausgabe.

Hinweise zum Betrieb mit FHEM

Der Selbstbau CUL wird fast genauso wie ein "richtiger" CUL verwendet, die entsprechende Dokumentation (http://fhem.de/commandref_DE.html#CUL) gilt also auch hier. Der einzige Unterschied ist der Bootloader. Der bereits bei fabrikneuen Arduinos vorhandene Bootloader ist nicht kompatibel zu dem Bootloader der culfw. Daher funktioniert das Kommando **B** zum rebooten des Selbstbau CULs nicht. Der Selbstbau CUL gerät dann in eine Endlosschleife, die sich nur durch aus- und einstecken beenden lässt. Das Problem tritt aber im normalen Betrieb nicht auf.

USB Geräte werden bei jedem Systemneustart neu *enumeriert*. Daher kann es vorkommen, dass ein Gerät, das mal unter /dev/ttyUSB0 erreichbar war nach dem Neustart unter /dev/ttyUSB1 ist. Das führt natürlich zu Problemen, wenn FHEM das Gerät unter /dev/ttyUSB0 erwartet. Um dieses Problem auszuschließen, empfiehlt es sich, den CUL unter seiner eindeutigen ID anzusprechen. Die entsprechende Gerätedatei findet sich unter /dev/serial/by-id.

Ein Beispiel für eine Definition in FHEM könnte so aussehen:

```
define nanoCUL CUL /dev/serial/by-id/usb-FTDI_FT232R_USB_UART_A903N5T5-if00-port0@38400 1234
```

Wichtig ist es, die Übertragungsrate von 38400 bit/s anzugeben.

Bekannte Probleme

Viele der Arduino nano haben eine sehr grell leuchtende blaue Stromversorgungs-LED. Diese lässt sich nicht per Software ausschalten. Sollte sie stören, kann entweder die Lötverbindung zur LED durchtrennt oder die LED z.B. mit einem schwarzen Lackstift übermalt werden.

Bei den nachgebauten nanos gibt es wohl viele mit einem Fehler (<http://forum.fhem.de/index.php/topic,24651.msg218291.html#msg218291>) in der Schaltung (unverbundener Testpin des FT232RL). Das kann dazu führen, dass er nicht richtig am USB Anschluss funktioniert.

Weblinks

- Blog [1] (<http://blog.gummibaer-tech.de/cul-stick-868433-im-selbstbau/>)
- Forumsthread (<http://forum.fhem.de/index.php/topic,24651.0.html>) zu diesem Thema
- Detaillierte Anleitung für Aufbau auf einer Lochstreifenplatine [2] (<http://raspberrypi.tips/raspberrypi-tutorials/hausautomatisierung-mit-fhem-teil-1-cul-stick-selbstbau-868mhz-cul-am-raspberry-pi/>)
- Weitere Konfigurationsoptionen auf der Seite über CUL

Abgerufen von „https://wiki.fhem.de/w/index.php?title=Selbstbau_CUL&oldid=18773“

Kategorien: Interfaces | CUL | Arduino

- Diese Seite wurde zuletzt am 24. Januar 2017 um 15:57 Uhr geändert.
- Der Inhalt ist verfügbar unter der Lizenz GNU-Lizenz für freie Dokumentation 1.3 oder höher, sofern nicht anders angegeben.