



f (<https://www.facebook.com/wolfgang.ewald.7503>)

@ (<https://www.instagram.com/wolfgang.ewald.7503/?hl=de>)




(<https://wolles-elektronikkiste.de/>)

Wolles Elektronikliste (<https://wolles-elektronikkiste.de/>)

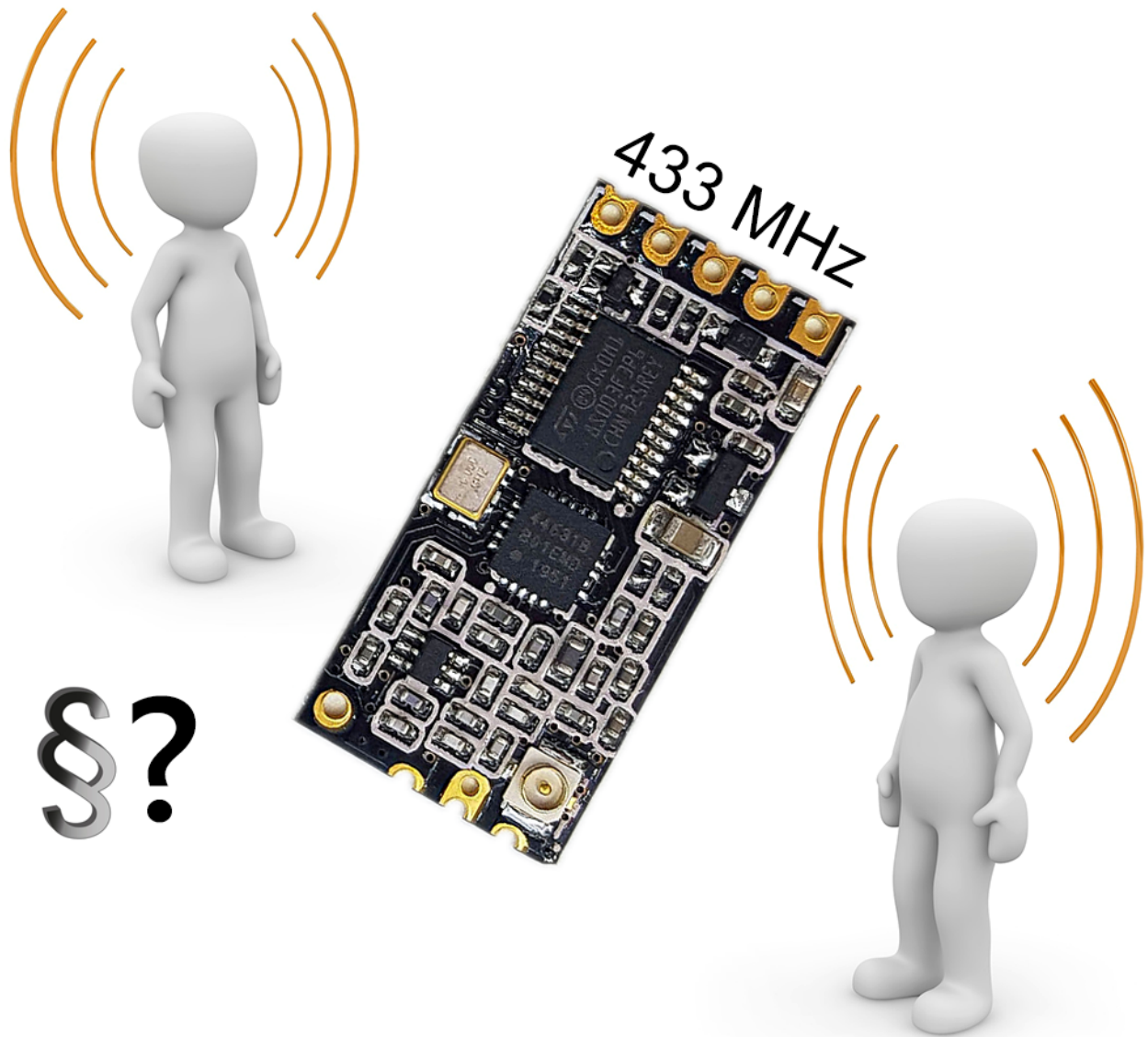
Die wunderbare Welt der Elektronik

ZUR STARTSEITE ([HTTPS://WOLLES-ELEKTRONIKKISTE.DE](https://wolles-elektronikkiste.de))

- ÜBER MICH UND DEN BLOG ([HTTPS://WOLLES-ELEKTRONIKKISTE.DE/UEBER-MICH-UND-DEN-BLOG](https://wolles-elektronikkiste.de/ueber-mich-und-den-blog))
- Beitragsverzeichnis (<https://wolles-elektronikkiste.de/beitragsverzeichnis>)
- Datenschutzerklärung (<https://wolles-elektronikkiste.de/privacy-policy>)
- BEITRAGSVERZEICHNIS ([HTTPS://WOLLES-ELEKTRONIKKISTE.DE/BEITRAGSVERZEICHNIS](https://wolles-elektronikkiste.de/beitragsverzeichnis))
- Ein paar Anmerkungen zur englischen Version (<https://wolles-elektronikkiste.de/ein-paar-anmerkungen-zur-englischen-version>)
- Impressum (<https://wolles-elektronikkiste.de/impressum>)
-  DEUTSCH ([HTTPS://WOLLES-ELEKTRONIKKISTE.DE/HC-12-FUNKMODUL](https://wolles-elektronikkiste.de/hc-12-funkmodul))
- Über mich und den Blog (<https://wolles-elektronikkiste.de/ueber-mich-und-den-blog>)

HC-12 Funkmodul

POSTED ON MÄRZ 12, 2021 ([HTTPS://WOLLES-ELEKTRONIKKISTE.DE/HC-12-FUNKMODUL](https://wolles-elektronikkiste.de/hc-12-funkmodul)) BY WOLFGANG EWALD ([HTTPS://WOLLES-ELEKTRONIKKISTE.DE/AUTHOR/WOLLES_EWOOD](https://wolles-elektronikkiste.de/author/wolles-ewald)).



Über den Beitrag

In einem meiner ersten Beiträge (<https://wolles-elektronikkiste.de/433-mhz-funk-mit-dem-arduino>) hatte ich schon einmal über 433 MHz Funkmodule berichtet, dabei aber das HC-12 Modul ausgelassen. Diese Lücke möchte ich nun schließen, zumal das HC-12 Modul einige sehr positive Eigenschaften hat. Es zeichnet sich zum einen durch überaus einfache Bedienbarkeit aus, zum anderen kann es sowohl als Sender wie auch als Empfänger eingesetzt werden.

Der Beitrag ist folgendermaßen gegliedert:

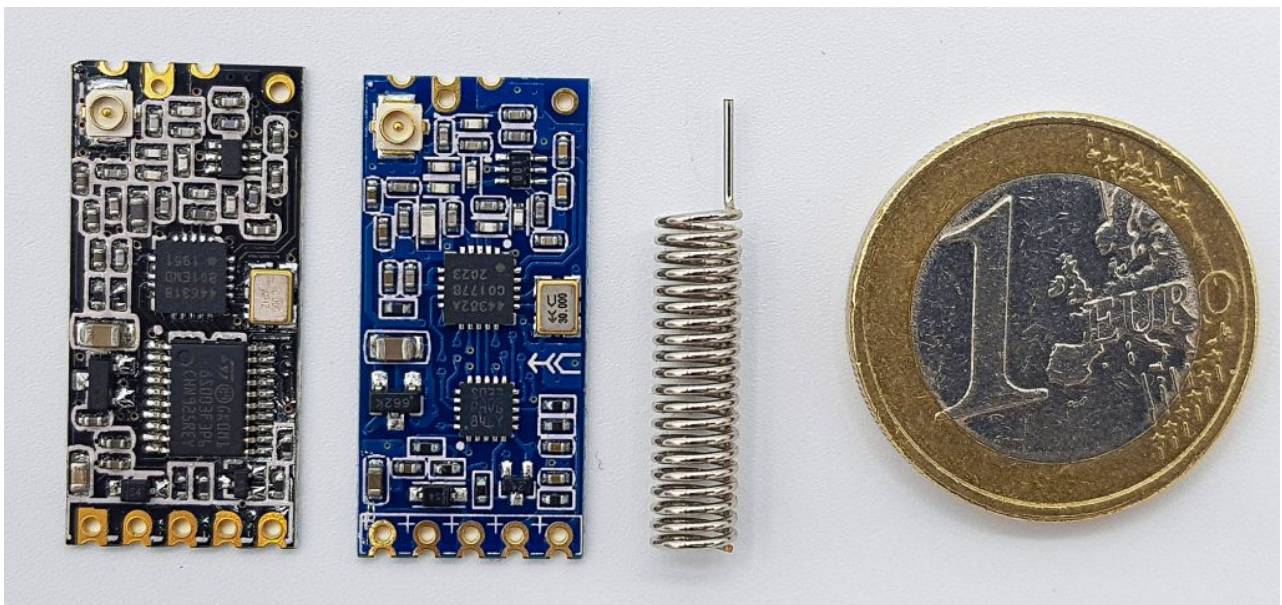
- Technische Eigenschaften
- Schnelleinstieg
- Einstellungen
- Reichweitentest

- Übermittlung von Sensordaten

Grundlagen / Technische Eigenschaften

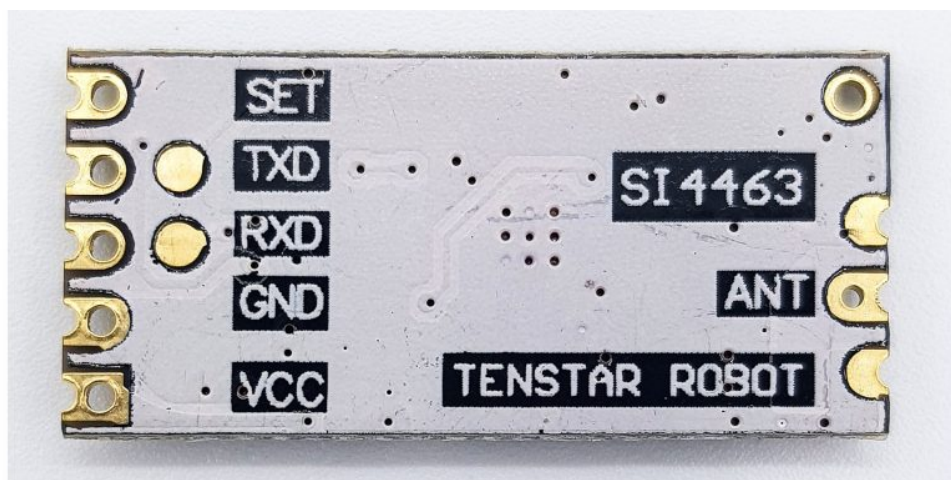
Das HC-12 Funkmodul bekommt ihr in Online-Shops für 6 bis 9 Euro. In chinesischen Shops ist es noch mal deutlich günstiger, ihr müsst dort aber ein paar Wochen Lieferzeit in Kauf nehmen. Normalerweise wird eine Spiralantenne mitgeliefert.

Das Moduldesign variiert. Die einen Module basieren auf dem SI4463 Transceiver, die anderen auf dem SI4438. Ersterer hat eine etwas höhere Reichweite (weitere Details hier (<https://www.iot-rf.com/si4438-rf-module-or-si4463-rf-module.html>)). Die Anschlüsse scheinen bei allen HC-12 Modulen gleich zu sein.



(<https://wolles-elektronikkiste.de/wp-content/uploads/2021/02/HC12.jpg>)

Zwei HC-12 Module mit Spiralantenne



(https://wolles-elektronikkiste.de/wp-content/uploads/2021/02/20210222_204455.jpg)

HC-12 Modul Rückseite

Schon mal vorab: nicht alles, was das HC-12 Modul in Bezug auf die Frequenzbereiche und Reichweiten kann, ist auch erlaubt (in Deutschland). Also nicht zu früh freuen. Ich komme darauf später noch zurück.

Hier die wichtigsten Daten im Überblick:

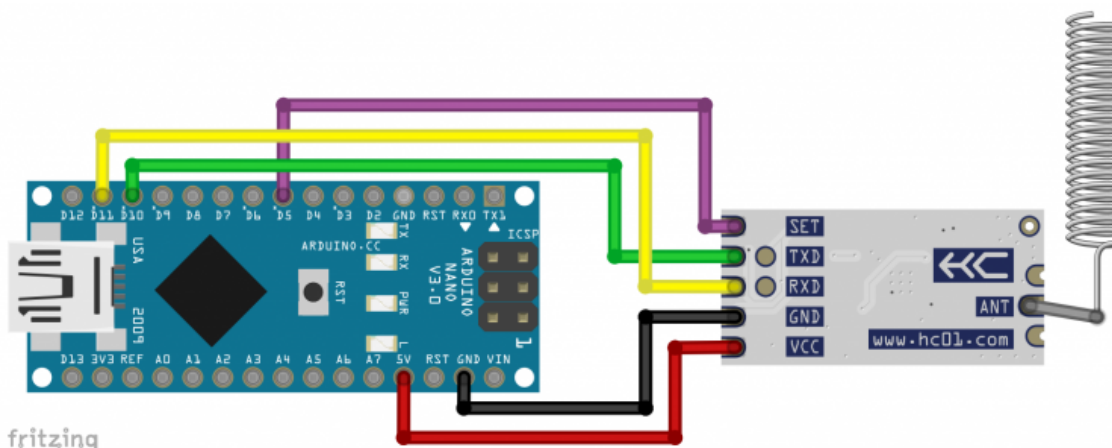
- Frequenzbereich: 433.4 – 473.0 MHz
- Strombedarf:
 - nicht sendend: je nach Betriebsmodus 80 μ A – 16 mA
 - beim Senden: ~100 mA
 - Schlafmodus: 23 μ A (eigene Messung)
- Spannungsversorgung: 3.2 – 5.5 Volt
- Kommunikation mit dem Modul: seriell (RX / TX)
 - Baudrate: 1200 – 115200
- Reichweite: maximal 1800 m
- maximale Sendeleistung: 20 dBm / 100 mW
- Übertragungsrate (Funk): bis 250000 bps
- Einstellungen über AT Kommandos (SET Pin)

Ein wesentliches Merkmal des HC-12 Moduls ist, dass es einen eigenen Microcontroller besitzt, der sich um die Funkübertragung kümmert. Ihr tretet nur über die serielle Verbindung mit dem Modul in Kontakt. Wie ihr sehen werdet, ist der Umgang mit den Modulen dadurch erstaunlich einfach. Ihr braucht euch mit keiner Funkbibliothek herumzuschlagen.

Schnelleinstieg

Schaltung

Lötet die Spiralantenne an das Modul und verbindet es mit dem Microcontroller oder Arduino eurer Wahl. Am Arduino Nano könnte es so aussehen:



(https://wolles-elektronikkiste.de/wp-content/uploads/2021/02/Wiring_HC_12.png)

Das HC-12 Modul am Arduino Nano

Das HC-12 Modul wird über Software Serial angesteuert. RX und TX des Moduls werden „über Kreuz“ mit den Software Serial Pins verbunden, d.h. RX an TX und TX an RX. Den SET Pin brauchen wir im Schnelleinstieg noch nicht, da wir die Standardeinstellungen nutzen.

Der Aufbau ist für die Sende- und die Empfängerseite identisch. Verbindet beide Arduinos mit dem Computer.

Der Basissketch

Als Einstieg nehmen wir den modifizierten SoftwareSerial Sketch aus den Beispielen der Arduino IDE. Wir wählen 9600 Baud, denn das ist die Voreinstellung des HC-12.

HC12_SoftwareSerial.ino

```
1.  #include <SoftwareSerial.h>
2.  SoftwareSerial hc12(10,11);
3.
4.  void setup() {
5.      Serial.begin(9600);
6.      hc12.begin(9600);
7.      Serial.println("Let's start!");
8.  }
9.
10. void loop() {
11.     if (hc12.available()) {
12.         Serial.write(hc12.read());
13.     }
14.     if (Serial.available()) {
15.         hc12.write(Serial.read());
16.     }
17. }
```

Ladet den Sketch auf den ersten Arduino, dann öffnet ihr eine zweite Instanz der Arduino IDE (also einfach die Arduino IDE noch mal aufrufen, ohne die erste zu schließen). Wählt den Port des zweiten Arduinos und ladet den Sketch auch auf den zweiten Arduino. Nun könnt ihr für beide einen seriellen Monitor öffnen. Schreibt eine Nachricht in eines der Eingabefelder, geht auf Senden und schon taucht die Nachricht im seriellen Monitor des anderen Arduinos auf.

(https://wolles-elektronikkiste.de/wp-content/uploads/2021/02/Simple_setup.png)

Ausgabe HC12_SoftwareSerial.ino

So einfach ist das! Einziges Problem: Wenn ihr in Deutschland wohnt, habt gerade gegen die gesetzlichen Bestimmungen zur Nutzung des 433 MHz Frequenzbandes verstoßen. Diese legen 10 mW als maximal zulässige Sendeleistung fest. Das HC-12 Modul sendet in seinen Standardeinstellungen jedoch mit 100 mW. Wir kommen gleich dazu, wie ihr in die Legalität zurückkehren könnt. Wenn ihr mehr über die Vorschriften für die Funknutzung wissen wollt, dann empfehle ich diesen Link (<https://www.mikrocontroller.net/articles/Allgemeinzuteilung>) zu einer leicht verständlichen Zusammenfassung.

Einstellungen am HC-12 Funkmodul

Die Einstellung erfolgt über AT Befehle, die ihr vielleicht schon von den HC-05 und HC-06 Bluetooth Modulen (<https://wolles-elektronikkiste.de/hc-05-und-hc-06-bluetooth-module>) kennt. Um in den AT Modus zu gelangen, müsst ihr einfach nur SET mit GND verbinden. Alternativ verbindet ihr SET mit einem I/O Pin und setzt diesen auf OUTPUT und LOW. Der SET Pin ist intern mit einem 10 kOhm Pull-Up Widerstand verbunden. Wenn ihr ihn wieder von GND trennt beziehungsweise am I/O Pin auf INPUT wechselt, geht er in den normalen Funkmodus zurück. Dafür braucht er ca. 40 Millisekunden.

Es macht einen Unterschied, ob ihr SET im laufenden Betrieb auf LOW zieht oder schon beim Einschalten des Moduls:

- Laufender Betrieb: das Modul bleibt weiterhin mit den für die serielle Verbindung gewählten Einstellungen (Baudrate, Datenbits, Paritätsbit, Stopbit)

erreichbar.

- SET an GND beim Einschalten: Die Baudrate für die Kommunikation mit dem Modul wird auf 9600 Baud gesetzt. Ferner gelten: 8 Datenbits, keine Parität, ein Stopp Bit.

Letztere Methode ist praktisch, falls ihr vergessen haben solltet, welche Baudrate ihr am Modul eingestellt habt. Sonst müsstet ihr schlicht durchprobieren.

Bringt das HC-12 Modul als ersten Test in den AT Modus. Auf dem Arduino nutzt ihr weiterhin den HC12_SoftwareSerial Sketch von oben. Tippt „AT“ (ohne die Anführungsstriche) in das Eingabefeld. Das Modul antwortet freundlicherweise mit „OK“.

Einstellung der Baudrate

Der folgenden Tabelle könnt ihr entnehmen, welche Baudraten ihr für die Kommunikation mit dem HC-12 Modul einstellen könnt. Die Funkbaudrate („Baud Rate in the Air“) wird von dem Modul automatisch angepasst.

(https://wolles-elektronikkiste.de/wp-content/uploads/2021/02/baudrate_serial_air_sensitivity.png)

Tabelle 1: Einstellung der Baudraten am HC-12 Modul

Eine höhere Datenrate erkaufte ihr euch mit einer geringeren Empfangssensitivität. Mit jeder Reduktion um 6 dBm verringert sich die Sensitivität auf die Hälfte.

Sende- und Empfangsmodul müssen die dieselbe Baudrate eingestellt haben. Die Einstellung erfolgt mit:

- AT+Bx mit x = 1200, 2400, 4800, usw.

Die Einstellung ist immer erst dann aktiv, wenn ihr den AT Modus verlassen habt.

Einstellung des Funkmodus

Das HC-12 Modul hat die vier Funkmodi: FU1, FU2, FU 3 (default) und FU4:

(https://wolles-elektronikkiste.de/wp-content/uploads/2021/02/HC12_Modi.png)

Tabelle 2: Die Funkmodi des HC-12 Moduls

Ihr stellt den Funkmodus mit der folgenden Anweisung ein:

- AT+FUx mit x = 1, 2, 3 oder 4

Funkmodus FU1

FU1 ist ein moderater Energiesparmodus. Damit verbraucht das HC-12 Modul 3.6 mA (3.8 mA nach meiner Messung) solange es nicht sendet. Ihr könnt in diesem Modus alle Baudraten wählen, allerdings ist die Funkbaudrate („in the air“) auf 250000 bps festgelegt. Das bedeutet, dass ihr eine hohe Datenrate habt, aber nur eine begrenzte Reichweite. Die angegebenen 100 m halte ich für sehr optimistisch. Außerdem reduziert sich dieser Wert, wenn ihr die Sendeleistung auf das in Deutschland erlaubte Niveau senkt.

Wenn ihr das Modul anweist etwas zu senden, dann geschieht das mit einer gewissen Verzögerung. Im Modus FU1 liegt die Verzögerung zwischen 15 und 25 ms.

Funkmodus FU2

Der Modus FU2 ist ein extremer Energiesparmodus. Der Verbrauch sinkt laut Datenblatt auf 80 µA. Ich habe sogar noch weniger gemessen, und zwar schwankte der Verbrauch periodisch zwischen 23 und 80 µA. Anscheinend geht das Modul zwischenzeitlich kurz in den Schlafmodus. Als Baudrate ist nur 1200, 2400 oder 4800 zulässig. Die Funkbaudrate ist mit 250000 bps wieder sehr hoch eingestellt und die Reichweite entsprechend begrenzt. Die Verzögerung beträgt 500 ms.

Falls ihr aus einem anderen Modus in FU2 wechselt und es war eine höhere Baudrate eingestellt, dann wird die Baudrate auf 4800 reduziert. Stellt ihr hingegen den FU2 Modus ein und wählt dann eine Baudrate größer 4800, funktioniert eure Übertragung nicht mehr.

Funkmodus FU3

FU3 ist der voreingestellte Modus. Alle Baudraten sind wählbar. Die Funkbaudrate wird entsprechend Tabelle 1 automatisch festgelegt. Der Verbrauch ist mit 16 mA recht hoch. Laut Datenblatt können 1000 m Reichweite erzielt werden (bei in

Deutschland unzulässigen 100 mW Sendeleistung). Die Verzögerung liegt – abhängig von der Baudrate – zwischen 4 und 80 ms.

Funkmodus FU4

Dieser Modus ist konzipiert, um hohe Reichweiten von bis zu 1800 Metern zu erreichen. Die einzig zulässige Baudrate ist 1200. Dabei wird die Funkbaudrate hier sogar auf 500 bps reduziert. Die Datenpakete sollten 60 Bytes nicht überschreiten und zwischen dem Senden zweier Pakete sollte eine Pause von 2 Sekunden liegen.

Einstellen der Sendeleistung

Jetzt kommen wir zu der angekündigten Einstellung, mit der ihr in den legalen Bereich der Sendeleistung kommt. Das HC-12 Modul bietet acht Stufen (Power Level):

(https://wolles-elektronikkiste.de/wp-content/uploads/2021/02/Power_Level_vs_Transmitting_Power.png)

Sendeleistung des HC-12 Moduls

Wie ihr unschwer erkennt, liegt die legale Grenze zwischen Level 4 und Level 5. Die Einstellung erfolgt über:

- AT+Px mit x = 1, 2, 3, 4, 5, 6, 7, 8

Die Voreinstellung ist 8.

Einstellung des Kanals

Ein schönes Feature des HC-12 Moduls ist die Möglichkeit 100 Kanäle einzustellen, deren Frequenzen um jeweils 400 kHz verschoben sind. Kanal 001 hat die Frequenz 433.4 MHz, Kanal 100 liegt bei 473.0 MHz. Leider gibt es hier wieder ein Problem (zumindest in Deutschland), denn der zulässige Frequenzbereich liegt zwischen 433.05 und 434.79. **Erlaubt sind damit nur die Kanäle 1 bis 4.**

Eingestellt werden die Kanäle mit:

- AT+Cx mit x = 001, 002, 003 099, 100

Die Voreinstellung ist Kanal 001.

Einstellung des Datenformats

Bei der seriellen Datenübertragung könnt ihr die Zahl der Bits einstellen, die direkt hintereinander übertragen werden. Etwas anderes als 8 wäre exotisch. Danach kommt ein Paritätsbit (<https://de.wikipedia.org/wiki/Parit%C3%A4tsbit>) – oder auch keins (non parity). Wird eines übertragen, könnt ihr wählen, ob auf gerade (even parity) oder ungerade Parität (odd parity) geprüft wird. Zum Schluss werden entweder 1, 2 oder 1.5 Stopp Bits übertragen.

Das Datenformat wird so eingestellt:

- AT+Uxyz mit:
 - x = Anzahl Bits, also normal 8
 - y = O (odd parity check), E (even parity check), N (no parity check)
 - z = 1 (ein Stop Bit), 2 (2 Stop Bits), 3 (1.5 Stop Bits)

Die Voreinstellung ist: 8N1.

Weitere Einstellungen

- AT+SLEEP versetzt das Modul in den Schlafmodus. Es benötigt dann lediglich 23 μ A. Der Schlafmodus beginnt, wenn ihr den AT Modus verlassen habt. Das Modul wird geweckt, indem ihr einmal in den AT Modus und wieder zurückgeht. Einen Beispielsketch dazu gibt es weiter unten.
- AT+DEFAULT setzt alle Einstellungen zurück.
- AT+UPDATE versetzt das Modul in einen Zustand, der das Aufspielen einer neuen Firmware erlaubt.

Abfragen des HC-12 Moduls

Ihr könnt nicht nur Einstellungen vornehmen, sondern diese auch abfragen:

- AT+V liefert die Firmware Version
- AT+Rx, mit
 - x = B, C, F oder P für Baud Rate, Kanal, Funkmodus oder Leistung
 - x = X bedeutet, dass alle Parameter abgefragt werden

So sieht dann die Antwort zu einer Abfrage mit „AT+RX“ aus:

(https://wolles-elektronikkiste.de/wp-content/uploads/2021/02/Output_AT_RX.png)

Abfrage des HC-12 Moduls mit AT+RX

Den Schlafmodus verwenden

Der Schlafmodus eignet sich besonders für Module, die nur ab und zu etwas senden sollen, wie zum Beispiel die Daten einer Wetterstation. Im Prinzip habe ich weiter oben schon erklärt, wie es funktioniert. Wichtig ist, dass an einigen Stellen Wartezeiten eingefügt werden.

Der Beispielsketch sorgt dafür, dass das HC-12 Modul alle 10 Sekunden ein freundliches „Hello again“ versendet und danach in den Schlaf geschickt wird.

HC12_Sleep.ino

```
1.  #include <SoftwareSerial.h>
2.  SoftwareSerial hc12(10,11);
3.  int SETPin = 5;
4.
5.  void setup() {
6.      pinMode(SETPin, INPUT);
7.      Serial.begin(9600);
8.      hc12.begin(9600);
9.      Serial.println("Let's start!");
10. }
11.
12. void loop() {
13.     hc12.print("Hello again");
14.     delay(100);
15.
16.     pinMode(SETPin, OUTPUT);
17.     delay(50);
18.     hc12.print("AT+SLEEP");
19.     delay(50);
20.     pinMode(SETPin, INPUT);
21.
22.     delay(10000);
23.
24.     pinMode(SETPin, OUTPUT);
25.     delay(100);
```

```
26.     pinMode(SETPin, INPUT);  
27.     delay(100);  
28. }
```

Reichweitentest

Um die Reichweite des Moduls zu testen, habe ich zunächst wieder die weiter oben abgebildete Schaltung verwendet. Die beiden Arduino Nano Boards habe ich mit einer 9 Volt Batterie betrieben. Die Sendeeinheit lief mit dem folgenden sehr einfachen Sketch, den ich wohl nicht erklären muss:

HC12_Distance_Test_Transmitter

```
1.  #include <SoftwareSerial.h>  
2.  SoftwareSerial hc12(10,11);  
3.  
4.  void setup() {  
5.      hc12.begin(9600);  
6.  }  
7.  
8.  void loop() {  
9.      hc12.print("Hi Wolle");  
10.     delay(3000);  
11. }
```

Auf der Empfängerseite habe ich dem Arduino Nano noch eine LED an Pin 6 spendiert. Der Empfängersketch prüft, ob eine Nachricht gesendet wurde. Ist das der Fall, dann prüft er, ob die Nachricht korrekt ist. Wenn das auch der Fall ist, leuchtet die LED für eine halbe Sekunde auf. Ich habe mich mit dem Empfänger vom Sender entfernt und geprüft, wann die LED aufhörte zu blinken oder dieses nicht mehr regelmäßig tat. Den Versuch habe ich auf freiem Feld mit direktem Sichtkontakt durchgeführt.

HC12_Distance_Test_Receiver

```
1.  SoftwareSerial hc12(10,11);  
2.  int ledPin = 6;  
3.  
4.  void setup() {  
5.      hc12.begin(9600);  
6.      pinMode(ledPin, OUTPUT);  
7.  }  
8.  
9.  void loop() {  
10.     String message = "";  
11.     if(hc12.available()) {  
12.         message = hc12.readString();  
13.         if(message=="Hi Wolle"){  
14.             digitalWrite(ledPin, HIGH);
```

```
15.         delay(500);  
16.         digitalWrite(ledPin, LOW);  
17.     }  
18. }  
19. }
```

Mit den Standardeinstellungen (FU3 / 9600 Baud / verbotene 100 mW Sendeleistung) und der mitgelieferten Spiralantenne konnte ich eine Strecke von ca. 120 Metern überbrücken. Das ist nur ein Bruchteil der weiter oben angegebenen 600 Meter.

Verbesserung der Reichweite

Durch zwei Änderungen der Hardware könnt ihr die Reichweite erhöhen:

- Stattet die Stromversorgung mit einem fetten Kondensator aus. Im Datenblatt werden 1000 μF empfohlen. Ich hatte nur 470 μF in meinem Fundus. Der Kondensator sorgt dafür, dass der plötzliche, hohe Strombedarf beim Senden gedeckt werden kann.
- Verwendet eine bessere Antenne. Aus früheren Versuchen (<https://wolles-elektronikkiste.de/433-mhz-funk-mit-dem-arduino>) weiß ich, dass ein 17.3 cm (= 1/4 der Wellenlänge) langer, gerader Draht schon bessere Ergebnisse liefert als die Spiralantennen.

Ich wollte jetzt aber aufs Ganze gehen und habe mir zwei externe 433 MHz Antennen für je ca. 15 Euro besorgt. Da diese einen SMA Anschluss hatten, musste ich mir noch zwei SMA auf IPEX Adapter besorgen. So sah der Aufbau aus:

(https://wolles-elektronikkiste.de/wp-content/uploads/2021/02/range_test.jpg)

Erhöhung der Reichweite der HC-12 Module mit externer Antenne

Und so als Schaltbild:

(https://wolles-elektronikkiste.de/wp-content/uploads/2021/02/HC12_Range_Test.png)

Fritzing Schema

Damit habe ich eine Reichweite von über 1000 Metern erzielt. Der limitierende Faktor war am Ende die freie Sicht. Ob tatsächlich 1800 Meter erreichbar sind, kann ich also nicht mit Bestimmtheit sagen.

Mir wurde erst dann bewusst, dass die eingesetzte Sendeleistung in Deutschland nicht erlaubt ist. Ich habe daraufhin weitere Versuche mit der legalen Sendeleistung (Powerlevel 4) durchgeführt. Bei Verwendung der Spiralantenne im Modus FU3 sank die Reichweite weniger als befürchtet auf 80 Meter. Durch den Wechsel auf FU4 ging es dann wieder hoch auf 180 Meter. Und mit der guten Antenne konnte ich dann 350 Meter erreichen.

Ich habe auch die Module mit dem SI4438 Chip getestet und konnte keinen Unterschied zu den SI4463 basierten Modulen feststellen. Allerdings habe ich nur einen Parametersatz ausprobiert.

Hier eine Zusammenfassung aller Testergebnisse:

(https://wolles-elektronikkiste.de/wp-content/uploads/2021/03/Range_Testergebnis.png)

Ergebnisse der HC-12 Reichweitentests

In meinem Haus konnte ich mit FU3 / 9600 Baud / Spiralantenne / Powerlevel 4 problemlos durch zwei Wände und eine Zimmerdecke hindurchsenden.

Schaut euch an, was ihr wirklich braucht und spielt ein bisschen herum. Und haltet euch an die gesetzlichen Vorgaben bezüglich der Sendeleistung. Ihr wollt schließlich auch nicht, dass eure 433 MHz Anwendungen durch andere gestört werden!

Übermittlung von Sensordaten

Als Abschluss noch – für die weniger Erfahrenen – eine Anleitung, wie ihr Sensordaten übertragen könnt. Genauer ausgedrückt geht es darum, wie ihr Integer und Float Daten als Strings zusammenfasst, sendet und auf der Empfängerseite wieder in Zahlen umwandelt.

Der Sender, der die Daten zur Verfügung stellt (Server) wandelt die Daten mittels `String(Zahl)` in Strings, die zusammengefügt und durch Nicht-Zahlen (hier Sternchen) voneinander getrennt werden. Erhält der Sender eine entsprechende Anfrage, wird der String gesendet. Der Sketch versendet feste Werte. Das würdet ihr durch Sensordaten oder Ähnliches ersetzen.

HC12_Data_Server.ino

```
1.  #include <SoftwareSerial.h>
2.  SoftwareSerial hc12(10,11);
3.
4.  void setup() {
5.      Serial.begin(9600);
6.      Serial.println("Let's start!");
7.      hc12.begin(9600);
8.  }
9.
10. void loop() {
11.     String requestString = "";
12.     if (hc12.available()) {
13.         requestString = hc12.readString();
14.         Serial.println(requestString);
15.     }
16.     if(requestString == "request"){
17.         String dataString = "";
18.         createDataString(dataString);
19.         Serial.println(dataString);
20.         hc12.print(dataString);
21.     }
22. }
23.
24. void createDataString(String &data){
25.     int sensor1 = 746;
26.     int sensor2 = 8295;
27.     float sensor3 = 41.84;
28.     data = "*" + String(sensor1) + "*" + String(sensor2) + "*" +
String(sensor3);
29. }
```

Auf der Empfängerseite (Client) habe ich einen Taster installiert, der die Anfrage auslöst.

(https://wolles-elektronikkiste.de/wp-content/uploads/2021/02/Wiring_HC_12_pushbutton.png)

Nach kurzer Zeit erhält der Client die Antwort vom Server. Um die Zahlen aus eingehenden String zu extrahieren, gibt es die überaus nützlichen Funktionen `parseInt()` und `parseFloat()`. Diese Funktionen analysieren die ankommenden Zeichen. Sie warten bis Zahlen auftreten, sammeln diese bis zur nächsten Nicht-Ziffer (außer dem Punkt bei `parseFloat()`) und wandeln die Zahlenkette in einen Integer- beziehungsweise Floatwert.

HC12_Data_Client.ino

```
1.  #include <SoftwareSerial.h>
2.  SoftwareSerial hc12(10,11);
3.  int requestPin = 7;
4.
5.  void setup() {
6.      pinMode(requestPin, INPUT);
7.      Serial.begin(9600);
8.      hc12.begin(9600);
9.      Serial.println("Let's start!");
10. }
11.
12. void loop() {
13.     int sensorData1 = 0;
14.     int sensorData2 = 0;
15.     float sensorData3 = 0.0;
16.
17.     if (hc12.available()) {
18.         sensorData1 = hc12.parseInt();
19.         sensorData2 = hc12.parseInt();
20.         sensorData3 = hc12.parseFloat();
21.
22.         Serial.print("Sensor 1: ");
23.         Serial.println(sensorData1);
24.         Serial.print("Sensor 2: ");
25.         Serial.println(sensorData2);
26.         Serial.print("Sensor 3: ");
27.         Serial.println(sensorData3);
28.     }
```

```
29.     if(digitalRead(requestPin)) {  
30.         delay(300); // debounce  
31.         hcl2.print("request");  
32.     }  
33. }
```

Und so sieht die Ausgabe auf der Sender- und der Empfängerseite aus:

(https://wolles-elektronikkiste.de/wp-content/uploads/2021/02/Output_Data_Transfer.png)

Ausgabe von HC12_Data_Server.ino / HC12_Data_Client.ino

Danksagung

Die beiden Männchen auf dem Beitragsbild stammen von Peggy und Marco Lachmann-Anke (https://pixabay.com/de/users/peggy_marco-1553824/?utm_source=link-attribution&utm_medium=referral&utm_campaign=image&utm_content=1015374). Das Paragrafensymbol habe ich Michael Schwarzenberger zu verdanken. Alles wie gewohnt von Pixabay (<https://pixabay.com/de/>).

Das HC-12 Modul als Fritzing Bauteil stammt von Estevão Trabbold (stvz) auf fritzing.org (<https://forum.fritzing.org/t/hc-12-module-433mhz-long-range-1-8km/1976>).

Posted in [433 MHz](https://wolles-elektronikkiste.de/category/kabellos/433-mhz) (<https://wolles-elektronikkiste.de/category/kabellos/433-mhz>), [Kabellos](https://wolles-elektronikkiste.de/category/kabellos) (<https://wolles-elektronikkiste.de/category/kabellos>). Tagged [433 MHz](https://wolles-elektronikkiste.de/tag/433-mhz) (<https://wolles-elektronikkiste.de/tag/433-mhz>), [AT Kommandos](https://wolles-elektronikkiste.de/tag/at-kommandos-2) (<https://wolles-elektronikkiste.de/tag/at-kommandos-2>), [Baudrate](https://wolles-elektronikkiste.de/tag/baudrate) (<https://wolles-elektronikkiste.de/tag/baudrate>), [Einstellungen](https://wolles-elektronikkiste.de/tag/einstellungen) (<https://wolles-elektronikkiste.de/tag/einstellungen>), [Funkmodi](https://wolles-elektronikkiste.de/tag/funkmodi) (<https://wolles-elektronikkiste.de/tag/funkmodi>), [Funkmodus](https://wolles-elektronikkiste.de/tag/funkmodus) (<https://wolles-elektronikkiste.de/tag/funkmodus>), [HC-12](https://wolles-elektronikkiste.de/tag/hc-12) (<https://wolles-elektronikkiste.de/tag/hc-12>), [Reichweite](https://wolles-elektronikkiste.de/tag/reichweite)

(<https://wolles-elektronikkiste.de/tag/reichweite>), [Schlafmodus \(https://wolles-elektronikkiste.de/tag/schlafmodus\)](https://wolles-elektronikkiste.de/tag/schlafmodus), [Technische Eigenschaften \(https://wolles-elektronikkiste.de/tag/technische-eigenschaften\)](https://wolles-elektronikkiste.de/tag/technische-eigenschaften)

← DCF77 – Funkuhr (<https://wolles-elektronikkiste.de/dcf77-funkuhr>)

MPU9250 – 9-Achsen Sensormodul – Teil 1 →
(<https://wolles-elektronikkiste.de/mpu9250-9-achsen-sensormodul-teil-1>)

30 thoughts on “HC-12 Funkmodul”

Olli N. (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-3464>)

Juli 16, 2021

Hallo Wolfgang,

ich schon wieder:) Wir hatten ja erst zu den RXB12 /RXB8 – Modulen geschrieben,der Tausch hat durcj Deine Hilfe problemlos geklappt, vielen lieben Dank.

Wenn Du Dich erinnerst, diese sitzen auf einer Relaisplatine (über Handsender angesteuert) .Wenn ich jetzt das RXB8 durch das HC-12 ersetzen wollen würde, was entspricht dem Data-Pin des RXB8 – TX, RX odetr Set?

Und sorry, wenn es eine blöde Frage ist im Vergleich zu den technischen Ausführungen oben, vor denen ich richtig Respekt habe 😊

Antworten

Wolfgang Ewald (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-3465>)

Juli 16, 2021

Die RXBx Module sind nicht mit den HC12-Modulen kompatibel. Die RXBx Module sind sehr simpel. Am Sender legst du Datenpin auf HIGH und solange sendet er ein Dauersignal. Am Empfänger dasselbe: so lange er was empfängt, geht sein Datenpin auf HIGH. Du kannst also die zu übertragenden Daten codieren, wie du willst. Normalerweise machen das dann Bibliotheken für dich, aber du könntest auch Morsezeichen senden oder die irgendetwas anderes ausdenken:

<https://wolles-elektronikkiste.de/eigene-funkprotolle> (<https://wolles-elektronikkiste.de/eigene-funkprotolle>)

Der HC-12 hingegen ist ein komplexes Bauteil mit eigenem Microcontroller. Du „siehst“ praktisch nur seine serielle Schnittstelle. Entsprechend übermittelst du deine Information bytewise. Der HC-12 übersetzt das dann entsprechend seinem Funkprotokoll. Auf der Empfängerseite wird das Funksignal dann rückübersetzt als Zeichenkette interpretiert und über die serielle Schnittstelle ausgegeben. Insofern ist der TX Pin der Datenausgang des HC-12 Empfängers, nur kommen da komplexe Daten heraus die dein Relais nicht versteht.

Antworten

Olli N. (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-3470>)

Juli 17, 2021

Danke für die ausführliche Antwort und schade, dann nützt mir das nix. Im Prinzip nutze ich es als Garagentor-Öffner mit einfachem Ein-Knopf-Handsender. Der Empfänger mit Spiralantenne sitzt innen 1m vom Tor und die Reichweite ist innen ok, ca. 40m.

Da es ein Metallgitter-Tor ist, ist die Reichweite nach außen aber nur ca. 3m, ich brauche aber mind. 8-10m. Habe schon ein Koaxialkabel nach draußen gelegt, aber egal, ob ich dieses als Antenne nutze oder eine andere Antenne zusätzlich dran hänge – nützt nix, die Reichweite bricht dann auch innen ein, obwohl ich an der Spiralantenne nichts ändere.

Ich werde einfach weiter probieren, es sei denn, du hast noch eine geniale Idee 😊

Gruß, Olli

Antworten

Wolfgang Ewald (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-3472>)

Juli 17, 2021

Theoretisch könntest du den das Empfangsmodul in einem wasserdichten Gehäuse nach draußen verlegen und das Datensignal dann per Kabel nach drinnen schicken.

Ich weiß nicht, was du an Antennen schon ausprobiert hast, ich habe mit dem im HC-12 Artikel beschriebenen Modell gute Erfahrungen gemacht. Das wäre vielleicht auch eine Option.

Mehr fällt mir an einfachen Dingen erst einmal nicht ein.

Antworten

Olli N. (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-3481>)

Juli 17, 2021

Es geht hier um eine Tiefgarage mit je 16 Parkplätzen auf jeder Seite, also ein ziemlich langer Schlauch. Das Tor wird über Schlüsselschalter geöffnet, was sehr lästig ist, wenn man im Auto sitzt, daher als Alternative die Funkschalter oder besser -taster.

Das Problem ist ja nicht die Reichweite an sich, sondern das „entweder-oder“. Entweder setze ich die Antenne nach innen ans Tor, dann deckt sie innen bis hinten alles ab, aber aussen nur zwei Meter... Das bringt nix, wenn man erst dreiviertel der Einfahrt runter muss (die auch noch stark gebogen ist), bevor das Tor reagiert.

Setze ich die Antenne nach außen, habe ich das gleiche Spiel, nur innen kaum Empfang. Hänge ich eine Antenne nach innen und eine zweite nach außen, halbieren sich gefühlt beide Reichweiten und das bringt erst recht nix.

Als Antennen verwende ich die kleinen Kupfer-Helix-Spiralantennen, habe es aber auch schon mit nur einem 17,3 cm Draht, einem zur Antenne umfunktionierten Koaxialkabel und

einem Selbstbau-Dipol versucht. Industrielle Antennen wie aus deinem Beitrag würde ich nach den bisherigen Erfahrungen nur kaufen, wenn ich wüsste, dass es dann wirklich funktioniert und ich nicht beidseitig wieder diesen Schwund habe.

Was ich nicht verstehe, dass diese ganzen LoRa Module in der Lage sind, komplexeste Daten zu übertragen, aber nicht fähig sind, so etwas simples auszuführen wie einen der zahlreichen Anschlüsse auf „high“ zu setzen, wenn ich am Universalsender (vorher im Relais angelernt) den Knopf drücke.... vorausgesetzt, ich habe Dich und das alles richtig verstanden 😊

Antworten

Joschy (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-3769>) August 11, 2021

Genau dieses Problem hatte ich auch.

Ich konnte das dann so lösen, dass ich in der Blechverkleidung zwischen Einfahrt und Ausfahrtspur einfach einen quadratischen

Ausschnitt von ca 18x8 cm in das Blech gemacht habe und dort auf der Garagenseite Empfänger mit der 17cm Drahtantenne platziert habe.

Aussen habe ich einfach aus dem Baumarkt so ein gelbes Plastikschild „Ausfahrt frei halten!“ darüber geklebt.

aus Plastik drüber gemacht.

Die Reichweiten sind nun beidseitig super.

Antworten

Eckard (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-3085>)

Juni 11, 2021

Hallo Wolfgang,

meine HC12 Module übertragen nur noch 25 Byte pro Paket im FU4 Modus Powerlevel 4.

Bei einem früherem Test gingen aber 60 Byte durch. Im FU3 Modus ist keine Verringerung feststellbar.

Sind dir derartige Verhaltensweisen bekannt?

Gruß Eckard!

Antworten

Wolfgang Ewald (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-3088>)

Juni 11, 2021

Hallo Eckard,

eigentlich sollten 60 Bytes funktionieren. Das ist jedenfalls die Angabe in einem dieser verbesserungswürdigen Datenblätter. Ich habe den Aspekt gar nicht überprüft. Aber komisch ist schon, wenn es vorher ging und jetzt nicht mehr. Ist denn ein tatsächlich ein SI4463 Chip auf deinen Modulen verbaut? Wobei das natürlich trotzdem nicht erklären würde, warum es auf einmal nicht mehr geht.

VG, Wolfgang

Antworten

Eckard (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-3098>)

Juni 12, 2021

Hallo Wolfgang,

hatte vergessen zu erwähnen, dass es sich hier um den 4438 handelt.

Einen Werksreset hatte ich auch durchgeführt. Ohne Erfolg.

Erschöpftes Datenvolumen 😞.

Danke für deine schnelle Antwort.

Gruß Eckard!

Antworten

Hub (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-3056>)

Juni 8, 2021

Hi Wolfgang,

da ist dir ein prima Beitrag gelungen. - In einem der im Netz umhersegelnden Datenblättern habe ich gefunden, daß das Modul nach einer Änderung der Betriebsparameter die geänderten Parameter abspeichert. Damit ständen sie beim nächsten Power-up wieder zur Verfügung und man könnte die Parameter im Vorab anpassen, bevor das Modul in die Anwendung gesetzt wird. Hast du das bei deinen Untersuchungen bestätigt gefunden?

Gruss, Hub

Antworten

Wolfgang Ewald (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-3057>)

Juni 8, 2021

Hallo Hub,

ja, das kann ich bestätigen. Die Einstellungen, die man per AT-Befehl vornimmt, bleiben erhalten. Das wäre sonst auch recht unhandlich.

VG, Wolfgang

Antworten

Tom (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-2251>)

April 8, 2021

Habe soeben per Zufall Deine Seite gefunden – großartig!
Eine Frage: ich wollte vor längerem schon etwas Basteln, um eine Benachrichtigung zu bekommen, wenn der Postbote am Briefkasten war – mit LoraWan bin ich leider gescheitert, da der Briefkasten aus Metall ist und das Signal nicht rauskommt .. meinst du das könnt mit dem Funkmodul klappen? Leider ist die Distanz relativ groß, der Briefkasten ist an der Nordseite des Hauses und ich wohne auf der Südseite in der 5. Etage (das Haus hat 5), sprich es müßte über das Flachdach drüberfunken ..

Danke + Gruß aus Köln,
Tom

Antworten

Wolfgang Ewald (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-2252>)

April 8, 2021

Hi Tom,

da kann ich dir leider keine großen Hoffnungen machen. So ein Metallgehäuse ist eine ideale Abschirmung. Schau zum Beispiel mal hier, da gibt es eine Tabelle mit Durchlässigkeitsdaten:

<https://www.tor7.de/funktechnik-grundlagen> (<https://www.tor7.de/funktechnik-grundlagen>)

Kannst du vielleicht die Antenne aus dem Briefkasten herausführen?

VG, Wolfgang

Antworten

Markus Kaiser (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-2229>)

April 5, 2021

Hallo Wolfgang,
deine Beiträge sind immer lückenlos und erstklassig! Sehr fundiertes Wissen, was du da mit uns teilst! Dafür ganz herzlichen Dank!

Viele Grüße

Markus

Antworten

Wolfgang Ewald (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-2231>)

April 5, 2021

Danke schön!

Antworten

Eckard (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-2145>)

März 30, 2021

Hallo,
hab da mal noch eine Anmerkung zum Aufbau.
Wäre es nicht günstiger den Elko direkt an das HC12 Modul zusetzen,
als den Nano zu puffern?

Gruß Eckard!

Antworten

Wolfgang Ewald (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-2152>)

März 30, 2021

Da magst du recht haben – ich würde sagen: einfach ausprobieren. Danke für den Denkanstoß!

Antworten

Werner (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-2133>)

März 29, 2021

Ich habe (leider) diese Internetsite vor Kurzem gefunden. Gratulation, ich finde sie super.
Zum Thema: also die Nutzungsbedingungen für die Frequenzbereiche kommen von der Bundesnetzagentur (www.bundesnetzagentur.de). Da gibt es u.A. ein PDF „Non-specific Short Range Devices“. Dem entnehme ich, dass es für 433 MHz keine zeitliche Beschränkung gibt.
... und gesund bleiben

Werner

Antworten

Wolfgang Ewald (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-2138>)

März 29, 2021

Vielen Dank!

Antworten

Eckard (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-2068>)

März 24, 2021

Hallo,

habe auch noch Erfahrungen mit den Modulen gesammelt.
Mit der Erkenntnis, dass das 4463 nicht mit dem 4438 spricht und auch umgekehrt.
Jedes Modul hat seine eigene Firmware. Hat jemand andere Erfahrungen gemacht?
Fazit: Bei Ausfall muss die gleiche Version beschafft werden oder gleich paarweise ersetzen.
Auf die Abbildungen bei dem Großversender A.. kann man sich nicht verlassen. Ich habe die andere Variante erhalten, als abgebildet.

Gruß Eckard!

Antworten

Wolfgang Ewald (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-2070>)

März 24, 2021

Meine Erfahrungen sind da anders, bei mir hat auch der 4463 mit dem 4438 „geredet“. Aber ich will nicht ausschließen, dass es vielleicht Module gibt, wo das nicht funktioniert. Wichtig ist, dass die Module dieselben Einstellungen haben (Funkmodus, Baudrate, Paritätsbits, usw.).

Antworten

Eckard (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-2071>)

März 24, 2021

Das mit den Einstellungen ist schon klar. Gleiche Module mit unterschiedlichen Einstellungen würden ja auch nicht miteinander kommunizieren.

Ich hatte zwei paare mit gleichen Einstellungen, überkreuz ging einfach nicht.

Wohl bemerkt, es waren nicht alle vier aktiv.

Mit der nächsten Lieferung versuche ich es noch einmal.

Gruß Eckard

Antworten

Thorsten Hartwig (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-2059>)

März 23, 2021

Das nenne ich mal eine solide Anleitung! Bei allen anderen Beschreibungen, die ich gefunden habe, war das eher auf dem Niveau: „Einschalten, kuck, es geht.“

Vor allem die in Deutschland bestehende Gesetzeslage nehme ich dankbar mit.

Antworten

Wolfgang Ewald (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-2062>)

März 23, 2021

Vielen Dank!

Antworten

Stefan (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-1930>)

März 14, 2021

Wieder einmal sehr schöner und klarer Artikel Wolfgang

Antworten

Helmut Holm (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-1889>)

März 13, 2021

@Koehne:

Die Begrenzung gilt im 868MHz Band, dass sie für 433MHz auch gilt ist mir neu, gibt es einen Link für diese Regelung?

@Wolle

Danke für diesen Beitrag, hatte schon früher mit diesem Platinchen gespielt und einfach Temp und Feuchte Werte in eine Hausautomation geschickt.

Gruß und kein Corona, Helmut

Antworten

Wolfgang Ewald (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-1891>)

März 13, 2021

Vielen Dank, da haben sich unsere Kommentare überschritten!

Antworten

Koehne (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-1886>)

März 13, 2021

Hallo,

eine sehr gute Übersicht über die Kommandos, ich habe schon andere Module verwendet, bei denen man

die erst mühsam durch Probieren heraus finden musste.

Noch ein ganz wichtiger Hinweis zu den gesetzlichen Regelungen:

Nicht nur die Leistung ist begrenzt, auch die Zeit: man darf nur etwa 1% der Zeit, also 3,6 Sekunden /Stunde

senden, das grenzt die Datenmenge/Stunde sehr ein!!!!

Für grössere Datenmengen sollte man auf 2,4 GHz ausweichen, da gibt es auch ähnliche Baugruppen,

die habe ich noch nicht verwendet.

Grüße

Bernhard

Antworten

Wolfgang Ewald (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-1890>)

März 13, 2021

Nach meinen Informationen gibt es für 433 MHz keine Beschränkungen der Sendedauer (duty cycle). Für z.B. 868 MHz aber schon. Wenn es da Regelungen gibt, von denen ich nichts weiß, dann wäre ich über einen Link dahin dankbar.

Antworten

Stephan (<https://wolles-elektronikkiste.de/hc-12-funkmodul#comment-2737>)

Mai 16, 2021

gemäß der PDF von der Bundesnetzagentur sehe ich das genauso, da steht unter „Zusätzliche Parameter / Frequenzzugangs- und Störungsminderungstechniken“ bei 433,050 – 434,790 MHz nichts.

Antworten

Schreibe einen Kommentar

Deine E-Mail-Adresse wird nicht veröffentlicht. Erforderliche Felder sind mit * markiert.

Kommentar

Name *

E-Mail *

Website

☐ Ja, informiere mich über neue Beiträge! Inform me about new posts! You will NOT be notified about new or follow-up comments!

Kommentar abschicken

ABO / SUBSCRIPTION?



Vorname

E-Mail *

Zur Datenschutzerklärung. (<http://wolles-elektronikkiste.de/privacy-policy>)

Abonniere / subscribe!

RECENT POSTS



○ Dehnungsmessstreifen (<https://wolles-elektronikkiste.de/dehnungsmessstreifen>)

○ ESP32 mit MicroPython programmieren (<https://wolles-elektronikkiste.de/esp32-mit-micropython-programmieren>)

○ MicroPython – Umstieg von Arduino (<https://wolles-elektronikkiste.de/micropython-umstieg-von-arduino>)

○ ICM-20948 9-Achsensensor Teil II (<https://wolles-elektronikkiste.de/icm-20948-9-achsensensor-teil-ii>)

○ ICM-20948 9-Achsensensor Teil I (<https://wolles-elektronikkiste.de/icm-20948-9-achsensensor-teil-i>)

RECENT COMMENTS



○ Wolfgang Ewald bei DS3231 – Echtzeituhr (<https://wolles-elektronikkiste.de/ds3231-echtzeituhr#comment-5791>)

○ Marco Kahlert bei DS3231 – Echtzeituhr (<https://wolles-elektronikkiste.de/ds3231-echtzeituhr#comment-5783>)

○ Wolfgang Ewald bei ATtiny85 / 84 / 45 / 44 / 25 / 24 programmieren (<https://wolles-elektronikkiste.de/attiny85-84-45-44-25-24-programmieren#comment-5777>)

○ Sergio bei ATtiny85 / 84 / 45 / 44 / 25 / 24 programmieren (<https://wolles-elektronikkiste.de/attiny85-84-45-44-25-24-programmieren#comment-5760>)

○ Wolfgang Ewald bei Dehnungsmessstreifen (<https://wolles-elektronikkiste.de/dehnungsmessstreifen#comment-5746>)

ARCHIVES



Monat auswählen ▼

THEMEN



○ Andere Bauteile (<https://wolles-elektronikkiste.de/category/andere-bauteile>)

○ Porterweiterung (<https://wolles-elektronikkiste.de/category/andere-bauteile/porterweiterung>)

○ Anderes (<https://wolles-elektronikkiste.de/category/anderes>)

○ Boards und Microcontroller (<https://wolles-elektronikkiste.de/category/boards-und-microcontroller>)

○ Kabellos (<https://wolles-elektronikkiste.de/category/kabellos>)

○ 433 MHz (<https://wolles-elektronikkiste.de/category/kabellos/433-mhz>)

○ Bluetooth (<https://wolles-elektronikkiste.de/category/kabellos/bluetooth>)

○ Infrarot (<https://wolles-elektronikkiste.de/category/kabellos/infrarot>)

○ WLAN (<https://wolles-elektronikkiste.de/category/kabellos/wlan>)

- Projekte (<https://wolles-elektronikkiste.de/category/projekte>)
- Sensoren (<https://wolles-elektronikkiste.de/category/sensoren>)
 - Abstand, Licht, Bewegung (<https://wolles-elektronikkiste.de/category/sensoren/abstand-licht-bewegung>)
 - Beschleunigung (<https://wolles-elektronikkiste.de/category/sensoren/beschleunigung>)
 - Strom, Spannung (<https://wolles-elektronikkiste.de/category/sensoren/strom-spannung>)
- Software und Tools (<https://wolles-elektronikkiste.de/category/software-und-tools>)
- Sound (<https://wolles-elektronikkiste.de/category/sound>)
- Unkategorisiert (<https://wolles-elektronikkiste.de/category/unkategorisiert>)
- Wireless (<https://wolles-elektronikkiste.de/category/wireless-2>)

FORMALES / FORMAL STUFF

- Datenschutzerklärung (https://wolles-elektronikkiste.de/?page_id=3)
- Impressum (https://wolles-elektronikkiste.de/?page_id=428)

ANDERES / OTHER STUFF

- Startseite (<https://wolles-elektronikkiste.de>)
- (<https://www.bloggerei.de>)  **BLOGGEREI** (<https://www.bloggerei.de/>)
- Raus in die Natur (-> zum Blog meiner Frau). (<https://meisengezwitscher-blog.de/>)
- Besucht mich auf Github (<https://github.com/wollewald>)

Designed by [Inkhive Web Design](https://inkhive.com/) (<https://inkhive.com/>). © 2021 Wolles Elektronikliste. All Rights Reserved.

Mehrsprachiges WordPress (<https://wpml.org/de/>) mit WPML