



Links:

[Intro](#), [Tilslutning til Arduino](#), [Tilslutning på Fumlebrædt](#), [Kodeeksempler](#), [Antenne, Modulation](#),

[Om IC-erne på boardet](#), [Mulige Baudrates](#),

[Avanceret setup af transmission-modes](#), [AT-Kommandoer](#),

Dokumentet er ikke færdigt endnu!! Redigeres løbende !!

Trådløs HC-12-UART

HC-12 kan bruges som en Wireless serial Port

Den kobles direkte på en UART, Arduinos pin 0 og 1 – hvis der ikke skal bruges USB-kommunikation, - eller på en SoftSerial port.

Den er bidirektionel, dvs. den kan både bruges som sender og som modtager. Dog ikke samtidig!! Det er jo samme antenne, der bruges. Det kaldes Half Duplex.

Rækkevidden er op til flere hundrede meter. Nogle steder er nævnt op til 2 km. Men det er jo som normalt afhængig af forholdene. Indendørs er det meget kortere, fordi der er så mange forhindringer for radiosignalet undervejs. Men fra vores værksted og op i kantinen kan sagtens lade sig gøre!!

Der medfølger en lille spole-antenne, der skal loddes i printet.

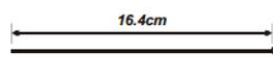
Alternativt kan der loddes en knap 17 cm. lang tråd i som antenne. ($\frac{1}{4}$ bølgelængde)

Spoleantennen genererer magnetiske svingninger, tråden elektriske



A. Helical antenna

0.5 mm enameled copper wire
close wound on 3.2 mm diameter former
433 MHz = 24 turns



C. Whip antenna

wire, rod, PCB-track or a combination
of these three

433 MHz = 16.4 cm total from RF pin.

Men om man starter med magnetfelt, - eller e-felt, er lige meget, da de genererer hinanden.

Efter sigende skulle en tråd-antenne række længere. Men ikke testet!

Men der kan også tilsluttes en ekstern SMA-antenne via et lille stik. Se senere



HC12 er default sat op til en Baud-rate på 9600 bps.

Men det kan man ændre i dens opsætning. Herom senere. Hold den bare på de 9600!! Det fungerer fint!!

HC-12 sender på ISM-båndet.

ISM-båndet – står for ” *Industrial, Scientific, and Medical Band* ” Det er et frekvensbånd der kan benyttes licensfrit til typegodkendt eller individuelt godkendt sendeudstyr og modtageudstyr.

Det ligger i området fra 433,05 til 434,79 MHz

Se: <https://da.wikipedia.org/wiki/ISM-b%C3%A5nd>

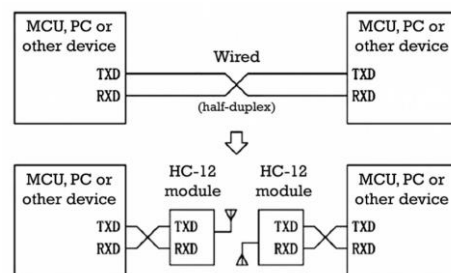
Måden, der sendes på er ved at sende korte Bursts, - hver gang der sendes en Byte, en pakke a´8 bit.

Her er vist en skitse af et transmissionssystem:

2 HC-12 moduler kan erstatte ledninger mellem 2 uC-er, som vist.

Bemærk, at i den øverste skitse, den wirede udgave mangler der en fælles Gnd.

Sender og modtager skal iflg. datablade være mindst 1,5 meter fra hinanden for at de virker!!



HC-12 koster ca. 120 kr., - men fås meget billigere i Kinesien.

- *Driftsspænding: 3.2V~5.5V*
- *Default address range (open field test): About 600m (maximum communication distance adjustable reach 1000m, the baud rate is 5000bps)*
- *Default idle current: 16MA (In different working modes operating current is different)*
- *Operating frequency range: 433.4-473.0MHz, up to 100 channels of communication*
- *The maximum transmit power: 100mW (settable)*
- *Default factory settings: Mode FU3, baud rate is 9600bps, communication channels CH001 (433.4M)*
- *Model: HC-12 433 SI4463*

HC-12 virker ikke sammen med forgængeren, HC-11.



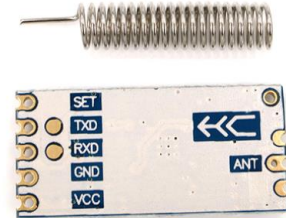
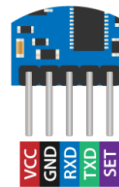
Tilslutning til Arduino

På HC12-boardet er der en microcontroller, der tager sig af data fra/til Arduinoen, og styrer selve radiosenderen – og modtageren, dvs. transceiveren.

HC-12 har 5 pins, Vcc, Gnd, Tx, Rx, og den sidste kan bruges til at indstille den til andre operatingmodes end default.

VCC og GND er hhv. 5 Volt og 0 Volt. Der bør monteres en kondensator på 22 uF eller mere mellem + og Gnd.

HC-12 Pinout



TxD (Transmit Data) sender data til den tilkoblede Arduinos RxD (Receive Data), - og RxD modtager data fra Arduino og sender dem ud via antennen.

HC12's strømforbrug i Bursts, - altså når der sendes, er 200 mA. Derfor bør den ikke strømforsynes fra USB-porten, men fra en separat 5 Volt supply.

Og der skal placeres en kondensator mellem 5V og 0 på mindst 22 uF, gerne 1000 uF. Plus også gerne en 100 nF.

Standby-forbruget er vist ikke ubetydelig. Måske kan det være en fordel at uC-en tænder for strømmen til HC-12 før der skal sendes. Det skal dog gøres på en bestemt måde. TxD og RxD på uC-en er normalt høje (NH). Så hvis man afbryder plus 5 Volt til HC-12, kan den "leve" (falsk) via TxD og eller RxD fra uC-en. Så den rigtige metode er at tænde / slukke Nul på HC-12 via en switch transistor.

Eller man kan få HC12 til at gå i dvale. Se senere !

Obs: Men har man sat HC12 i dvale, eller afbrudt dens nul, kan den jo ikke modtage !!

Nogle steder tilrådes at montere en 1N4007 diode i serie med dens Ucc-pin hvis den forsynes med mere end 4,5 Volt. Men det har jeg nu aldrig gjort!!

Arduinoens UART har indbygget en seriel buffer på 64 byte til både sende – og modtage-delen. Så det bør man holde sig under. Alternativt kan data " gå tabt".

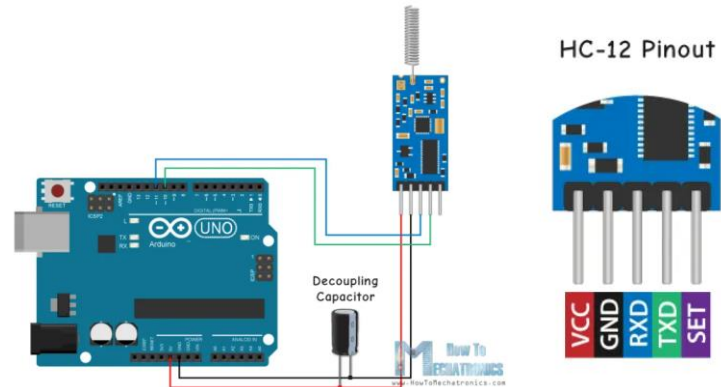
Der kan bruges flere modtagere / sendere i et system, blot de arbejder på samme Baudrate og ikke sender samtidigt.

Tilslutning af HC-12



Tilslutning til Arduino Uno.

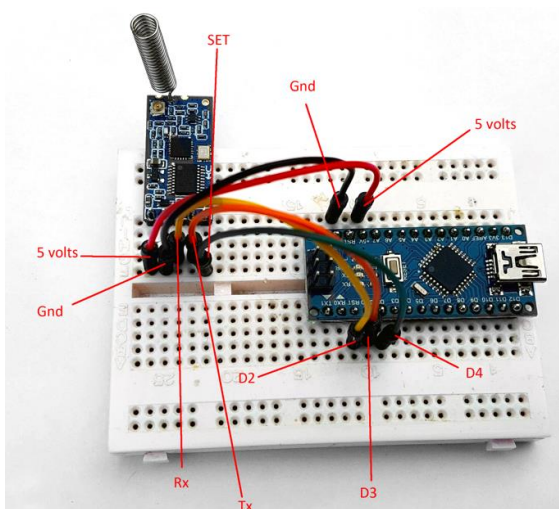
Bemærk, Set kan udelades. Den skal blot svæve.



Kilde: <https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-12-long-range-wireless-communication-module/>

Eksempel på opbygning på Fumlebrædt:

HC12's SET-pin skal kun bruges hvis man vil ændre på dens indstillingerne. Man kan bare undlade at bruge den, - lade den svæve !!



Her er HC-12 sat i et fumlebrædt !

Kilde:
<https://rydepier.wordpress.com/2017/02/03/hc-12-433mhz-radio/>

Kodeeksempel

Her er gengivet et kodeeksempel:

```
// Kodeeksempler

/* HC12 Send/Receive Example Program 1
   By Mark J. Hughes
   for AllAboutCircuits.com
```



Connect HC12 "RXD" pin to Arduino Digital Pin 4
Connect HC12 "TXD" pin to Arduino Digital Pin 5
Connect HC12 "Set" pin to Arduino Digital Pin 6

Do not power over USB. Per datasheet,
power HC12 with a supply of at least 100 mA with
a 22 uF - 1000 uF reservoir capacitor.
Upload code to two Arduinos connected to two computers.

Transceivers must be at least several meters apart to work.

```
*/  
#include <SoftwareSerial.h>  
const byte HC12RxdPin = 4;           // Recieve Pin on HC12  
const byte HC12TxdPin = 5;          // Transmit Pin on HC12  
SoftwareSerial HC12(HC12TxdPin,HC12RxdPin); // Create Software Serial Port  
  
void setup() {  
  Serial.begin(9600);                // Open serial port to computer  
  HC12.begin(9600);                 // Open serial port to HC12  
} // endsetup  
  
void loop() {  
  if(HC12.available()){              // If Arduino's HC12 rx buffer has  
data  
    Serial.write(HC12.read());        // Send the data to the computer  
  }  
  if(Serial.available()){            // If Arduino's computer rx buffer  
has data  
    HC12.write(Serial.read());        // Send that data to serial  
  }  
} // Endloop  
  
// Fra <https://www.allaboutcircuits.com/projects/understanding-and-implementing-the-hc-12-wireless-transceiver-module/>
```

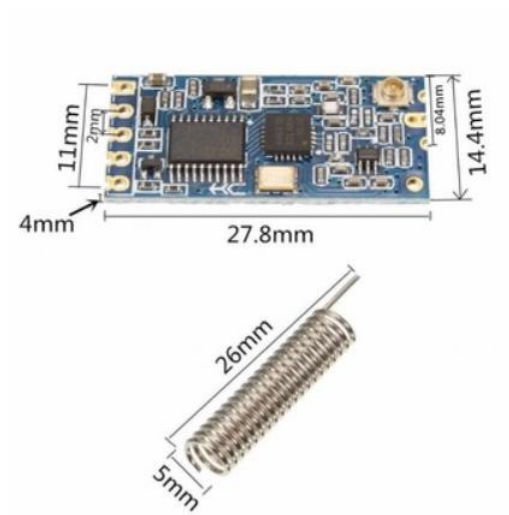
Kodeksempel # 2

```
/* Arduino Long Range Wireless Communication using HC-12  
Example 01  
by Dejan Nedelkovski, www.HowToMechatronics.com  
*/  
  
#include <SoftwareSerial.h>  
  
SoftwareSerial HC12(10, 11); // HC-12 TX Pin, HC-12 RX Pin
```



```
void setup() {  
  Serial.begin(9600); // Serial port to computer  
  HC12.begin(9600); // Serial port to HC12  
} // Endsetup  
  
void loop() {  
  while (HC12.available()) { // If HC-12 has data  
    Serial.write(HC12.read()); // Send the data to Serial monitor  
  }  
  while (Serial.available()) { // If Serial monitor has data  
    HC12.write(Serial.read()); // Send that data to HC-12  
  }  
} // Endloop
```

Spring antenna or SMA antenna



Der følger en spole-antenne til HC-12 til at lodde i boardet. Den fylder kun lidt, men er ikke den bedste antenneytpe!



Men der kan også monteres en ekstern antenne.

På printet er der også et stik, der passer til en SMA-antenne.



Eksempel på en SMA-antenne



Og et stik, der passer på HC-12, kabel og chassis-stik der passer med en antenne.

Fra Wiki:



SMA (SubMiniature version A) connectors are semi-precision coaxial RF connectors developed in the 1960s as a minimal connector interface for coaxial cable with a screw-type coupling mechanism.

Her ses stikket forstørret



Der fås lidt forskellige typer. Det kan godt forvirre !!

IPX IPEX to SMA Female Jack U.FL RF Pigtail Antenna Cable Fra [ArduinoTech](#)

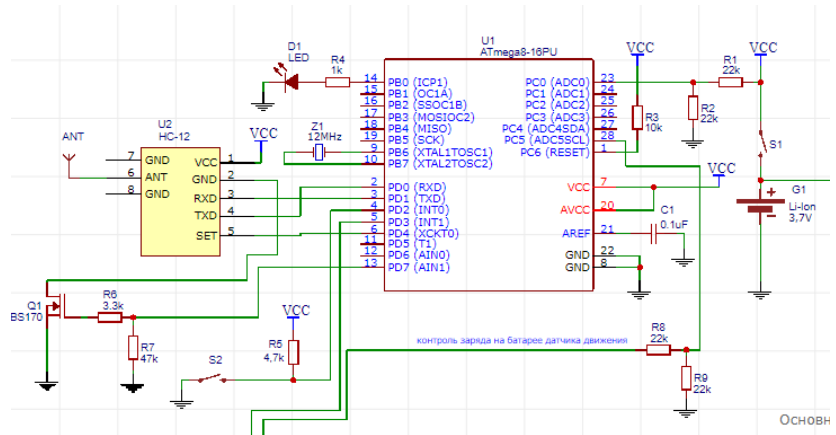
Tænd / sluk for HC-12



HC-12's standbyforbrug er vist ikke ubetydelig.

Diagram taget fra EasyEDA.

Her ses hvordan man kan afbryde Gnd på HC-12 printet. Der er brugt en lille MOSFet transistor. En BC337 kan sagtens bruges !!



Modulation

HC-12 433MHz FSK RF

The 434Mhz Serial RF Module HC-11 (1-40M) serial RF module is a low cost, high performance transparent FSK transceiver with operating at 434 MHz

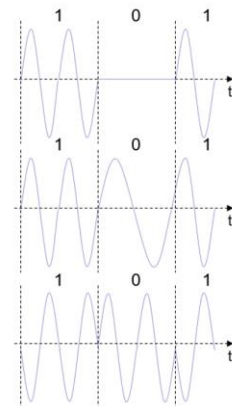
■ Over-the-air data rate (Example)

- Assume a remote unit needs to send 1000 bytes of payload data in a response to a 2-byte access point command every 75 milliseconds:
 - D_{ao} would be 32 bits and D_{ro} 80 bits
 - The total amount of data for both transmissions must occur in 75 milliseconds

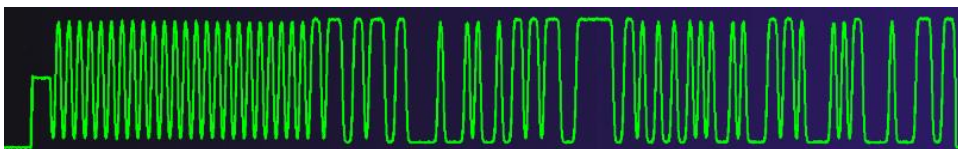
$$\text{RF data rate} = \frac{(((16b+32b+80b)+(8000b+32b+80b)) \times 1.1) / 0.075\text{sec}}{= 120.853 \text{ Kb/sec}}$$

■ Modulation Techniques

- **Amplitude Shift Keying (ASK):**
 - very simple
 - low bandwidth requirements
 - very susceptible to interference
- **Frequency Shift Keying (FSK):**
 - needs larger bandwidth
- **Phase Shift Keying (PSK):**
 - more complex
 - robust against interference



<http://www.dartmouth.edu/~engs112/presentations/RF-for-WirelessComm-Jose.ppt>



Advanced setup

AT-kommandoer



HC-12 indeholder en uC, der kan programmeres vha. såkaldte "AT"-kommandoer.

Når HC-12 er forbundet til Uno-ens UART, eller Softserial, kan man – ligesom man vil sende data til radiosending – sende Setup-kommandoer til den. Det kræver blot, at dens Set-pin er gjort lav.

Vha. af AT-kommandoerne kan man ændre indstillinger, Baudrate, Sendekanal mm, - men man kan også tjekke om kommunikationen til HC-12 virker. Ligeledes kan man bede om at få udskrevet hvordan HC-12 er indstillet.

HC-12 har en default opsætning ab fabrik.

Men der er rig mulighed for at ændre dette. Det gøres ved første at trække SET lav, og så sende nogle AT-kommandoer til den.

Dette kan gøres i et program, men også direkte fra debug-vinduet.

Eks:

Upload flg. kode I Arduinoen, der er koblet til HC-12:

```
#include <SoftwareSerial.h>

SoftwareSerial HC12(2, 3); // RX, TX. ( RX til Tx på HC12, og .. )

void setup() {
  Serial.begin(9600);
  HC12.begin(9600);
}

void loop() {
  while (HC12.available()) { // If HC-12 has data
    Serial.write(HC12.read()); // Send the data to Serial monitor
  }
  while (Serial.available()) { // If Serial monitor has data
    HC12.write(Serial.read()); // Send that data to HC-12
  }
}
```

Obs:

Efter Set er trukket lav, skal der ventes mindst 40 ms før der sendes AT-kommandoer.

Og efter set igen er gjort høj, skal den ha mindst 80ms. til at få indstillingerne på plads.

Kommunikationen fra Debugvinduet skal være indstillet til 9600 og Ny linje skal slås fra.



Setpin skal være lav, evt. styret fra en Ardu-pin!

AT-kommando:	Forklaring, indstil eller tjeck indstilling	Svar fra HC-12 i debugvinduet
AT	Test om der er forbindelse til HC-12 i AT-Mode.	OK
AT+RX	<u>Tjek</u> alle "common" Parametre Mode, Channel, Baudrate & Power	Viser indstillingerne
AT+DEFAULT	<u>Indstil</u> til default mode	OK+ ??
AT+Bxxxx AT+B4800	<u>Indstil</u> Baudrate (default = 9600) 1200, 2400 .. *1	OK+4800
AT+Cxxx AT+C021	<u>Indstil</u> kanal(001 = 433,4 MHz) 001 til 127 (400 KHz Spacing) Brug helst >5 kanaler imellem	COK+C021
AT+FUx	<u>Indstil</u> Function Uart (1 til 4) Sender og modtager skal være ens *2	AT+OK
AT+Px	<u>Set</u> Transmit Power, 1 til 8 8 = 20 dBm, (~100 mW) 1 = 0.8mW *3	OK+RP:+20dBm
AT+R?	<u>Tjeck</u> HC-12's indstillinger: ? = B, Baudrate ? = C, Channel ? = F, Function ? = P, Transmitting Power	OK+9600
AT+Udps AT+U801	<u>Indstil</u> databit, paritetsbit og stopbit Default U801	OK+U801
AT+V	<u>Tjek</u> Firmware Version	HC-12_V1.1
AT+SLEEP	<u>Sæt</u> HC-12 i Sleep-mode. (22 uA) For at få en sovende HC-12 til at vågne op igen, skal dens SET-pin kortvarigt trækkes lav, og derefter igen sættes høj. Se *4	OK+SLEEP



--	--	--

***1**

Rækkevidde

Rækkevidden bliver lavere ved højere baudrate.

1200 & 2400bps ~1000m,
4800&9600bps ~500m,
19200&38400bps ~250m,
57600&115200bps ~100m.

***2, Mode**

FU1

Moderate power saving mode with 250000bps “over the air” baud rate. Serial port baud rate can be set to any supported value.
Standby strøm: 3,6 mA

FU2

Extreme power saving mode with 250000bps “over the air” speed. Serial port rate is limited to 1200bps, 2400bps, 4800bps
Standby strøm: 80 uA

FU3 (default)

Har kun et delay på 4 – 80 ms.
Standby strøm: 16mA

FU4

Long range mode. (kun vers. >= 2.3). Har et delay på 1 sek.
Send Max 60 bytes med intervaller på mindst 2 sek.
Sender kun på 1200bps.

***3**

Følsomhed

Receiver sensitivity differs from -117dBm to -100dBm, depending on transmission speed.

***4**

Sleep mode

Pseudokode:

Sove

SET-pin = lav
Vent 25 ms
Send ”AT+SLEEP”
Efter respons, sæt SET-pin høj.

Vågne igen:

SET-pin trækkes lav
Vent 25 ms
SET-pin sættes høj igen
Vent 250 ms



Kodeeksempel på AT-kommandoer via et program

```
/*
 * All codes by Milan Karakas, http://wildlab.org
 * So far, I use library "SoftwareSerial.h", but later it will be changed
 * with manual UART protocol to allow me various tricks and more reliable
 * work. This code(s) are protected by fictive "beerware union", where you may
 * use all codes, change it, copy, sell, exchange, etc. In other words, no
 * any copyright, but no warranty that it will work properly as well, so as-is.
 */

#include <SoftwareSerial.h>

SoftwareSerial mySerial(4, 5); // RX, TX
//Remember that TX of the HC-12 goes to RX of the Arduino board,
//and RX of the HC-12 goes to the TX of the same board!
float volt;
int P1, P2, P3, P4;
byte packet[20];
int j;

void setup()
{
  pinMode(A6, INPUT); //just a reminder - it is input by default
  pinMode(6, OUTPUT); //set mode AT command, connected to RX of the HC-12 module
  pinMode(13, OUTPUT); //debug LED onboard Arduino nano or Arduino pro mini
  Serial.begin(9600);
  mySerial.begin(9600);
  analogReference(EXTERNAL); //not used exactly for displaying status of the local battery.
    //it is planed to use with resistive voltage divider 2:1
    //and 3.3 V input on "REF" pin on Arduino nano, but on
    //Arduino pro mini, such pin does not exist, or at least
    //it is not on pin header - require delicate soldering
    //on board... If used "default" option, then pay attention
    //to 5V power supply - if in error (for example 4.8V instead 5V),
    //then voltage readings may be wrong as well

  while (!mySerial);
  digitalWrite(6,0);
  delay(280);
  mySerial.print("AT+B9600");
  delay(40);
  if (mySerial.read() != 79) digitalWrite(13,1); // if first letter is not "O" from HC-12 feedback "OK+B9600"
  //Serial.println("Error setting speed of 9600 bps ");
  else digitalWrite(13,0);
  // Serial.println("Uart speed is 9600 ");
  for (int i=0;i<12;i++) mySerial.read(); //just read 12 bytes to flush out RX buffer
  delay(40);
  mySerial.print("AT+C001");
  delay(40);
  if (mySerial.read() != 79) digitalWrite(13,1); // if first letter is not "O" from HC-12 feedback "OK+C001"
  //Serial.println("Error setting channel");
  else digitalWrite(13,0);
  //Serial.println("Channel is 001 - 433.400 MHz ");
  for (int i=0;i<12;i++) mySerial.read(); //just read 12 bytes to flush out RX buffer
  delay(40);
  mySerial.print("AT+P5");
  /* P1=0.8 mW
   * P2=1.6 mW
   * P3=3.2 mW
   * P4=6.3 mW
   * P5=12.6 mW
```



```
* P6=25.1 mW
* P7=50 mW
* P8=100 mW
*/
delay(40);
if (mySerial.read() != 79) digitalWrite(13,1);// if first letter is not "O" from HC-12 feedback "OK+P1"
//Serial.println("Greska kanala");
else digitalWrite(13,0);
//Serial.println("Power is 12.6 mW "); //only if P5 is set, your choice.
//For indoor testing, please set P1 or 0.8 mW, then for field you may test at higher power
//Pay attention to good antennas on TX, to prevent damage! 0.8 mW is "error free" even with
//shitty spring antennas, but those antennas are small and handy - just for testings

for (int i=0;i<12;i++) mySerial.read(); //just read 12 bytes to flush out RX buffer
  delay(80);
  mySerial.print("AT+U8N2");
  delay(80);
if (mySerial.read() != 79) //digitalWrite(13,1);// if first letter is not "O" from HC-12 feedback "OK+U8N2"
Serial.println("Errpr setting 1 startbit, 8 data bits, no parity, and 2 stop bits");
else digitalWrite(13,0);
//Serial.println("8 bits, no parity, 2 stop bits ");
delay(80);
delay(40);
digitalWrite(6,1);
delay(80);
}
```

Kilder:

<https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-12-long-range-wireless-communication-module/>

<http://www.instructables.com/id/Long-Range-18km-Arduino-to-Arduino-Wireless-Commun/>

<https://www.quora.com/How-can-I-send-data-over-the-HC12-RF-module-using-Arduino>

Datablad, PDF: https://arduinoshoppen.dk/help/HC-12_User_Manual.pdf

Link: <https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-12-long-range-wireless-communication-module/>

<https://www.instructables.com/id/Long-Range-18km-Arduino-to-Arduino-Wireless-Commun/>

<https://www.makerlab-electronics.com/product/433mhz-serial-rf-module-hc-12-1000m/>

<https://picaxeforum.co.uk/threads/hc-11-and-hc-12-transceiver-modules.28893/>

<http://www.electronicfreak.net/hc-12-module-testing-code/>

<http://www.14core.com/wiring-the-hc11-hc12-434433mhz-transceiver/>



Beskrivelse af koden: se: <https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-12-long-range-wireless-communication-module/>

Kilder: <https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-12-long-range-wireless-communication-module/>

Fix dårlige kloner: <https://www.youtube.com/watch?v=ZfBuEAH-Q8Y&t=167s>

Om IC-erne på boardet

The Si4463 Transceiver

The Si4463 provides the wireless communication in this circuit. It has a maximum transmit power of 20 dBm (100 mW) and receive sensitivity of -129 dBm. Two 64-byte Rx and Tx FIFO memories are built into the chip along with a great many advanced features that are not implemented in the HC-12 design. See the datasheet for more information on multiband operation, frequency hopping, etc.

The STM8S003FS Microcontroller

This is an 8-bit microcontroller with 8 kB of flash memory, 128 bytes of EEPROM, and a 10-bit ADC. It supports UART, SPI, and I²C and has multiple I/O pins. It offers many of the same capabilities as its ATmega and XMC counterparts. It is programmed to control the Si4463 as well as handle the UART communication between the HC-12 and whatever it is connected to on the other end.

The HC-12 Transceiver Module

Combined with other components, the Si4463 and STM8S003 create the HC-12 transceiver, which provides a 4-pin TTL-level UART interface (Vcc, Gnd, Tx, Rx), with a 5th pin that is used to enter "command" mode for changing the module's configuration. The HC-12 has 100 supported channels spaced 400 kHz apart, eight transmit levels, eight supported baud rates, and three different working modes.

The 5th pin on the HC-12 is labeled "Set" and, when driven to logic low, allows various settings to be selected on the HC-12 using AT commands sent to the "RXD" pin.

The default configuration of the HC-12 is FU3—on Channel 1, FU3 is a fully automatic and transparent (to other devices) setting that adapts to the transmission rate of the connected device (although 9600 baud is still required to program it in Command mode).



Note that as the transmission rate increases, the sensitivity of the receiver decreases. You can return to the default state by sending AT+DEFAULT once in command mode.

Mulige Baudrates

Serial Port Baud Rate	Over-the-Air Baud Rate	Receiver Sensitivity
1200 bps	5000 bps	-117 dBm
2400 bps	5000 bps	-117 dBm
4800 bps	15000 bps	-112 dBm
9600 bps	15000 bps	-112 dBm
19200 bps	58000 bps	-107 dBm
38400 bps	58000 bps	-107 dBm
57600 bps	236000 bps	-100 dBm
115200 bps	236000 bps	-100 dBm