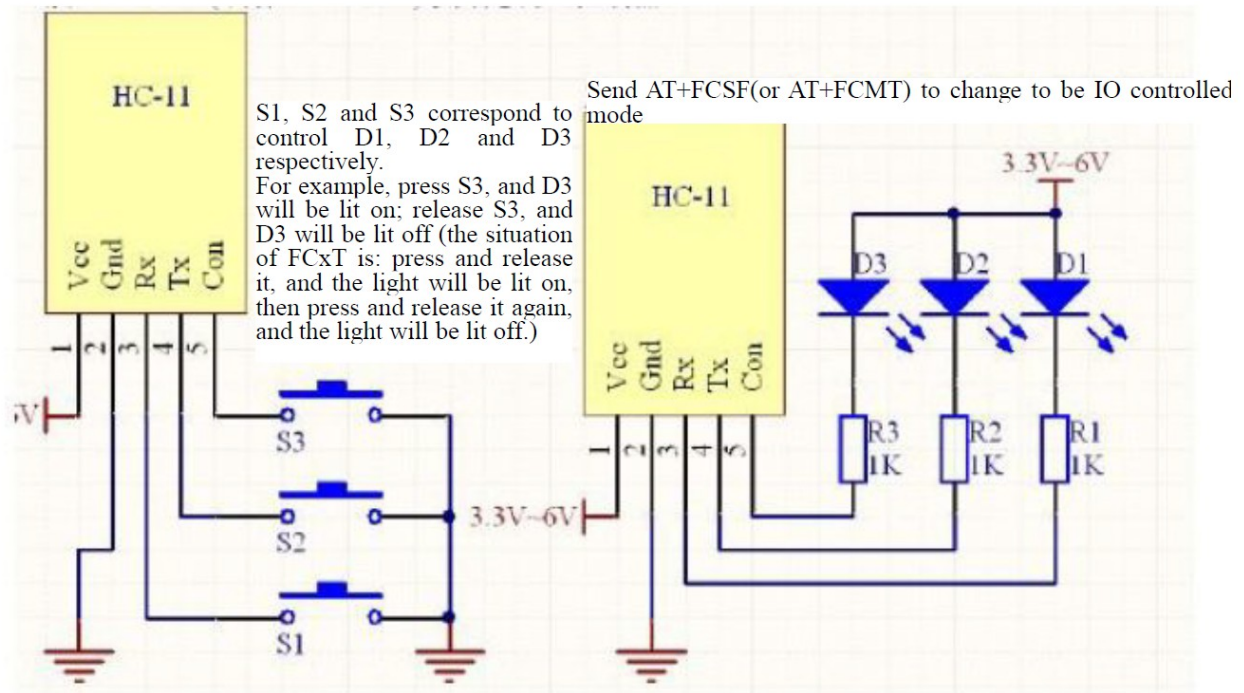


HC11/12-433MHz Transceiver

Mit diesem Baustein hatte ich viele Probleme. Allein die Testschaltung aus dem sog. Datenblatt:

Figure IV. Simple Test of Wireless IO Control

Send AT+FCMF (or AT+FCMT) to change to be wireless IO controller



gibt zu denken. Die Dioden(LEDs) sind so ungewöhnlich eingezeichnet, dass ich zunächst den oberen Sammelanschluss auf Masse legte. Nichts tat sich! Aber dann bemerkte ich, dass der Controller überhaupt nicht sendete.

Mit dem Spektrumanalysator beobachtete ich das Signal an der Antenne. Da kam nichts raus! Dann schrieb ich erneut ein kleines Programm mit dem Arduino und testete am Oszilloskop die serielle Schnittstelle TX auf PIN5 bzw. RX am HC11. Es kamen offensichtlich die verlangten Signale am Baustein an, wenn ich mit dem Reset das Programm startete. Nun war guter Rat teuer. Denn der Baustein sollte eigentlich im AT-Mode korrekt gesetzt sein. Links der Sender(Controller), rechts der Empfänger. Die Codes waren richtig gesetzt. Was nun?

Programm

Ich schrieb mein Konfigurationsprogramm um und beobachtete erneut am Oszillografen die seriellen Signale, die zum RX-Eingang des Bausteins gingen. Der Serielle Monitor der Arduino-Entwicklungsumgebung zeigte folgende vom Baustein zurückgegebenen Zustände und eine Warnung (respons = ERROR) an.

```
73   HC11ByteIn = HC11.read();
74
75
76   Serial.print(HC11ByteIn);
77
78   }
79
80   //Serial.print("0D");
81   delay(200);
82
83
84   //AT+FCSF = Empfänger(HC11 as receiver),
85
86   HC11.write("AT+FCMT");
87
88
89   delay(80);
90
91   while (HC11.available()) {
92     HC11ByteIn = HC11.read();
93
94     if (HC11.available()){
95       Serial.print(HC11ByteIn);
96     }
97   }
98
```

```
OK+C015
OK+B1200
ERROR
```

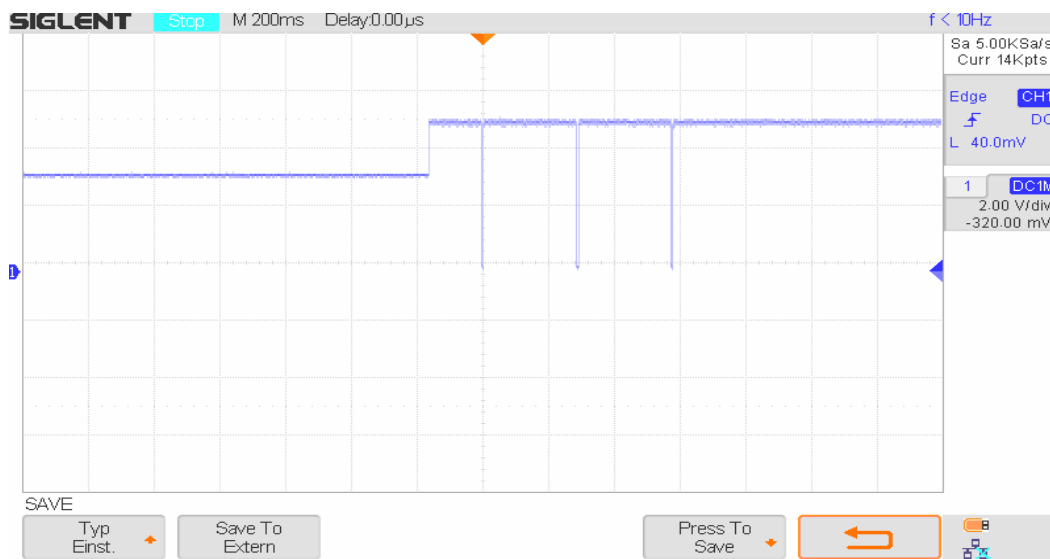
Nur beim Setzen von AT+FCMT kam also die Fehlermeldung: **ERROR!**

Nun kann es aber sein, dass der Baustein im AT-Mode darauf nicht antwortet. Also obwohl im Datenblatt erwähnt, keine Quittierung abgibt. Jedenfalls hatten ihn die Signale erreicht(Oszillograf) und ich ging davon aus, dass dieser Baustein jetzt als Controller sendet.

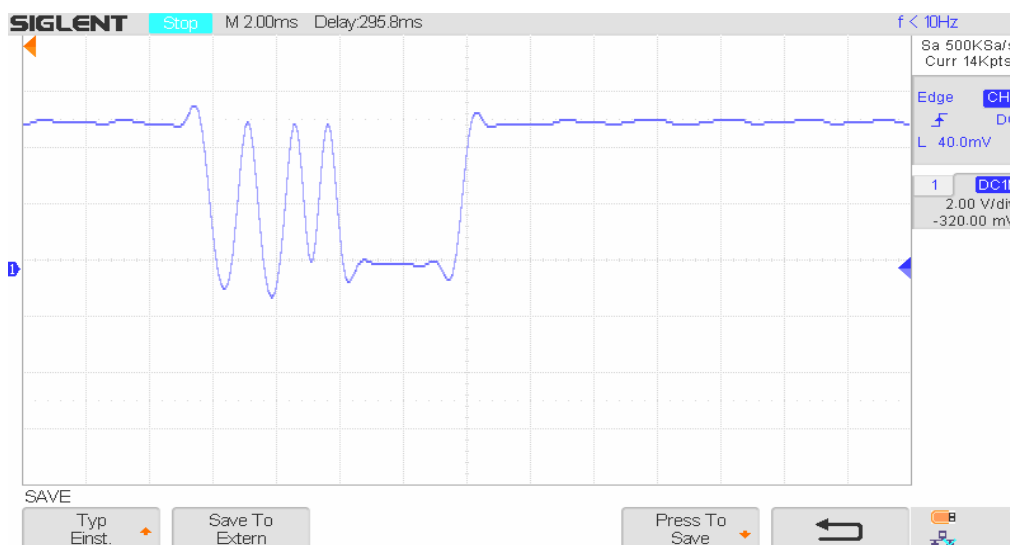
Danach flashte ich den zweiten Baustein als remote controlled receiver mit AT+FCST. Und natürlich zeigte sich der gleiche Zustand inkl. des Errors. Die beiden Bausteine stellte ich > 50cm nebeneinander und tastete am Sender, um die LEDs am Empfänger zu schalten. Nichts passierte!

Screenshots

Hier der Nachweis, dass Signale vom Arduino-UNO zum Baustein gingen:



Alle drei Schreibphasen kamen durch. Aber was sind das für Impulse?



Trotz eines Stützkondensators 330uF am Baustein, schaltete der Arduino offensichtlich nicht korrekt durch. Der Tastkopf des Oszilloskops war auf DC eingestellt! Ich versorgte den Arduino-UNO zusätzlich mit einer externen Gleichstromquelle und machte eine neue Aufnahme. An der Form der Impulse änderte sich leider nichts. Ich prüfte den Tastkopf am Kalibrier-Anschluss: Die Impulse waren schön rechteckig.

Nun, der Hersteller verrät, dass da im Inneren stets 1k in Reihe geschaltet sind. Und außerdem geht er vom TTL-Pegel aus, wenn man ordentlichen Betrieb machen will. Kann ja sein, doch zunächst blieb ich bei 3,3V DC. Denn einen Baustein hatte ich schon zerschossen, er gibt keine Antworten mehr.

Was tun?

Nachbrenner

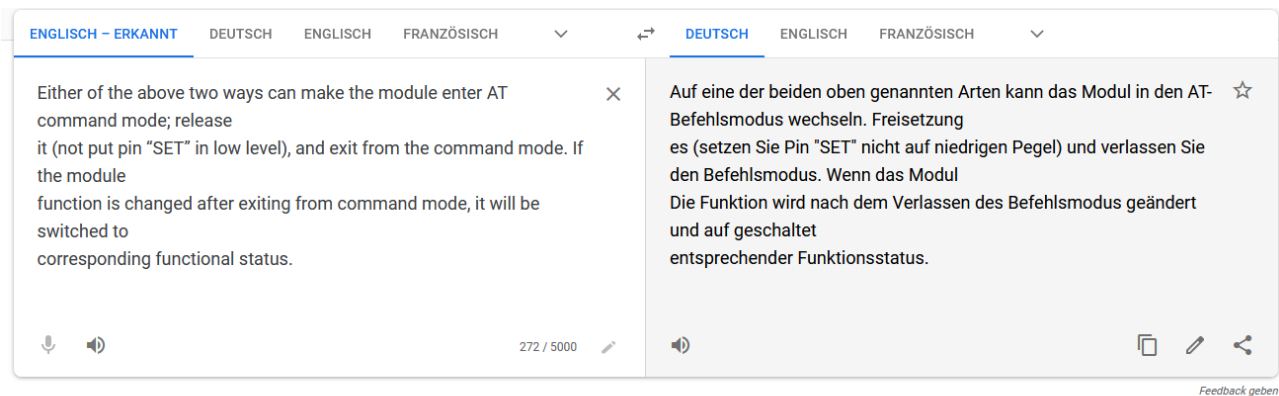
Wenn der Arduino an seinen Datenausgängen nicht schnell genug Ladung liefern kann, dann muss eine Schaltung her, die das macht. Also bastelte ich einen Treiber mit Transistoren.

The default setting is FU3, 9,600bps (8-dibit data, no check, one stop bit), CH001 (433.4MHz).

Sollte ich vielleicht diese Einstellung setzen? Dabei wird an anderer Stelle empfohlen, die Baudrate auf 1200 einzustellen, damit man eine große Entfernung mit wenigen Fehlern überbrücken kann. Denn ich möchte ja einen Gleichstrommotor möglichst ohne Aussetzer in zwei Drehrichtungen ansteuern. Dafür genügen Relais. Eventuell kann ich mit einem dritten Kanal die Geschwindigkeit(Drehzahl) verändern.

Datenblätter

Sie verdienen den Namen nicht. Habe aber nochmal nachgelesen:



Und wie bitte soll man aus diesem Kauderwelsch schlau werden?

Ja was ist denn nun eigentlich der Zustand nach dem Flashen?

Die aus dem Chinesischen nach Englisch übersetzten „Datenblätter“ sind für mich leider unverständlich. Warum kann denn ein Entwickler den Text nicht mal in einfacher Sprache verständlich darstellen? Da haben wir Internet und solche Verständigungsprobleme! Kann ich nicht nachvollziehen.

Man findet nur wenige wirklich hilfreiche Artikel im Netz. Die meisten sind von Anbietern, die nur verkaufen wollen. Und bei den brauchbaren Texten machen die Verfasser keine Bemerkung zu Problemen. Bei denen funzt das alles sofort. Da kann ich mich nur wundern. Aber ich habe Geduld. Inzwischen argwöhnte ich einen Fehler in den Beschreibungen. Es scheint so, dass der HC11 nicht den AT+FCSF versteht. Das kann aber der HC12, wie geschrieben steht. Soll ich mir jetzt auch noch zwei HC12 zulegen?

Machte mich doch zunächst an den Treiber. Denn ich traute den Impulsformen nicht, selbst wenn manchmal plausible Antworten zurück kamen. Ein erfahrener Elektroniker will saubere Signale sehen!

Neue Messungen

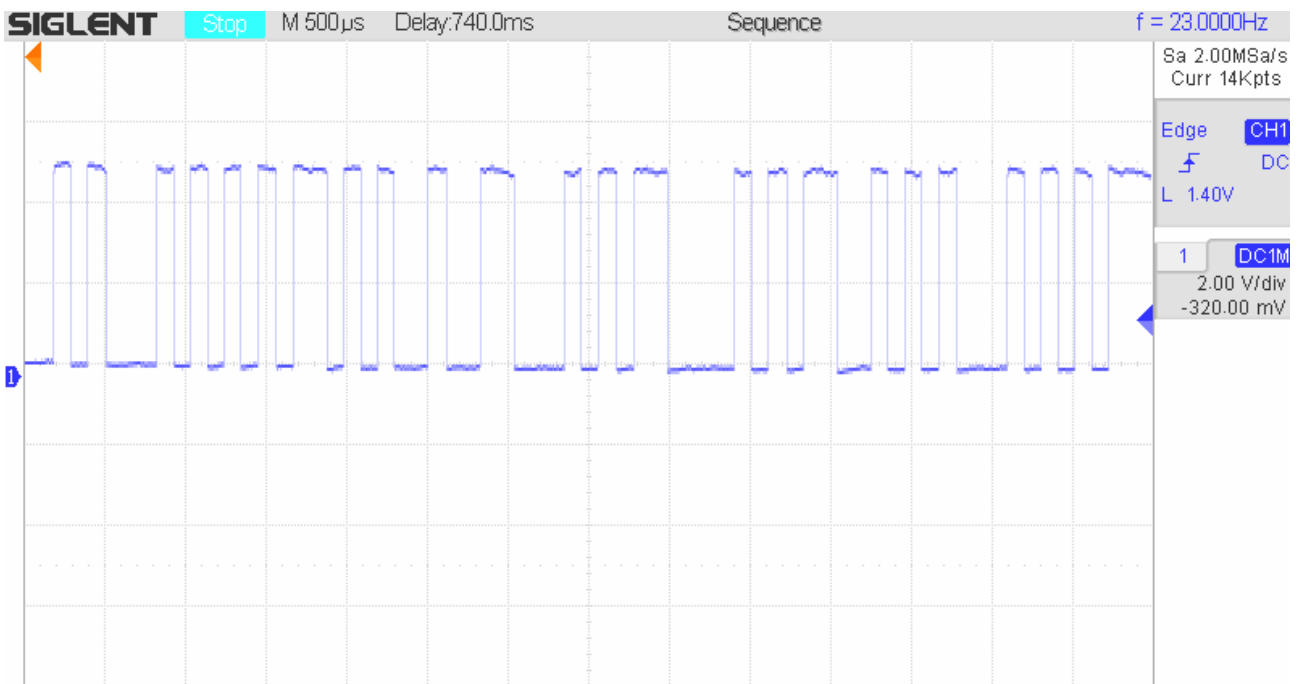
Transistor mit 47 Ohm Kollektorwiderstand:



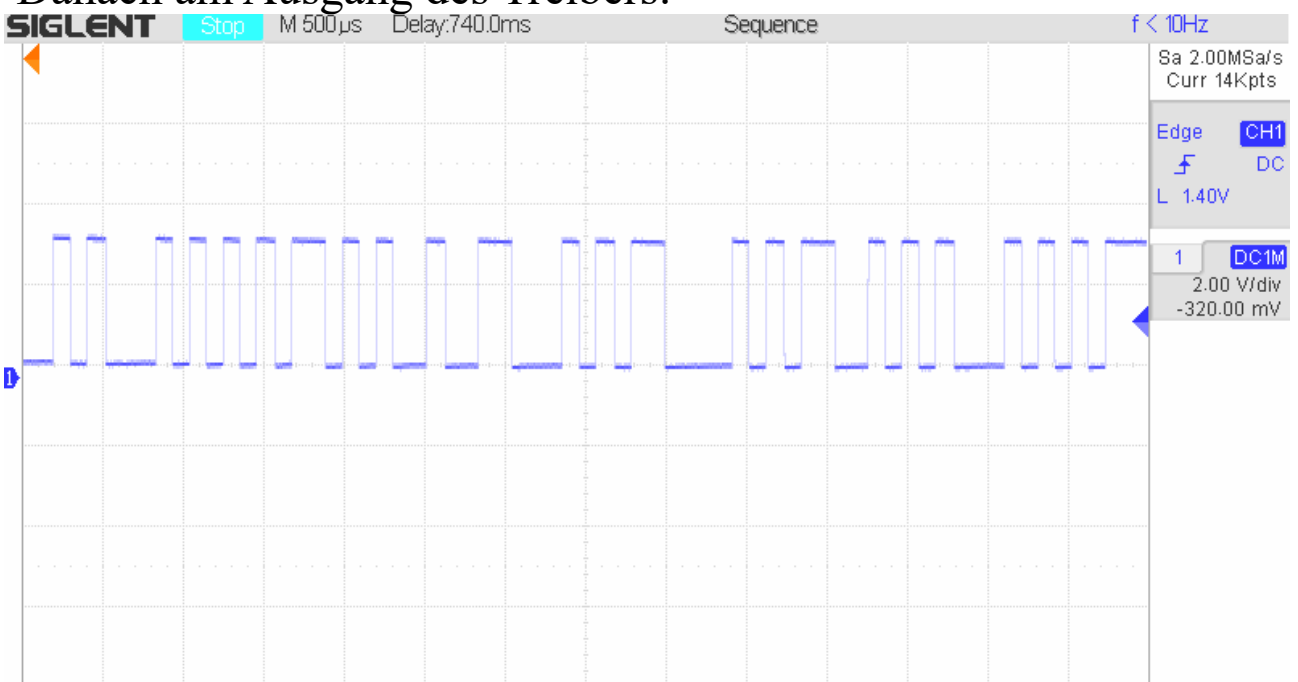
Die Impulse am RX-Pin des HC11 haben sich nicht verbessert! Was ist da los? Hat da vielleicht der Aufzeichnungsspeicher des Siglent einen Einfluss? Ich weiß es nicht. Jedenfalls kommt immer noch der ERROR bei AT+FCST oder AT+FCMT, nachdem ich über einen zweiten Transistor die richtige Phasenlage eingestellt hatte! Also doch einen HC12 kaufen?

Man lernt nie aus

Und es lag doch am Oszillografen! Erstmals musste ich mich mit dem Aquisition-Mode auseinandersetzen. Nach dem Einschalten dieses Betriebs zeigte sich die wahre Ursache für die Verformung der Rechteckimpulse. Im ersten Bild am TX-Ausgang des UNOs, im zweiten nach dem Treiber. Da die Signale am Ausgang des UNOs eigentlich ganz brauchbar sind, könnte ich den Treiber weglassen. Der Aufbau auf einem Breadboard war mir sowieso zu unsicher. Nun kam der soeben eingetroffene HC12 ins Spiel.



Danach am Ausgang des Treibers:



SEQUENCE

Acq. Mode
On

Segments Set
1

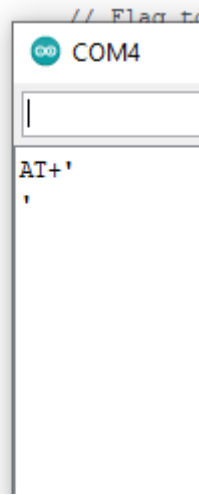


Die Amplituden(5Vss) der Signale am TX sind höher als nach dem Treiber. Lasse ich nicht doch lieber den Treiber in der Schaltung? Er liefert immerhin korrekt 3,3Vss. Also aus Sicherheitsgründen eben doch mit Treiber! Ich wagte den ersten Programmierversuch an dem HC12.

Was zeigt der Monitor?

Kommt da eine Antwort bei AT+FCSF? Nee, denkste:

```
103 SerialEnd = false;
104 HC11End = false;
105
106
107
108 //Serial.println("AT+FCMF");
109
110 HC11.write("AT+FCSF");
111 while (HC11.available()) {
112     HC11ByteIn = HC11.read();
113     HC11ReadBuffer += char(HC11ByteIn);
114     if (HC11ByteIn == '\n') {
115         HC11End = true;
116     }
117 }
```



Was ich auch an Konfigurationsdaten eingab, es kam immer dieselbe Response: 'A+'. Immerhin kein ERROR!

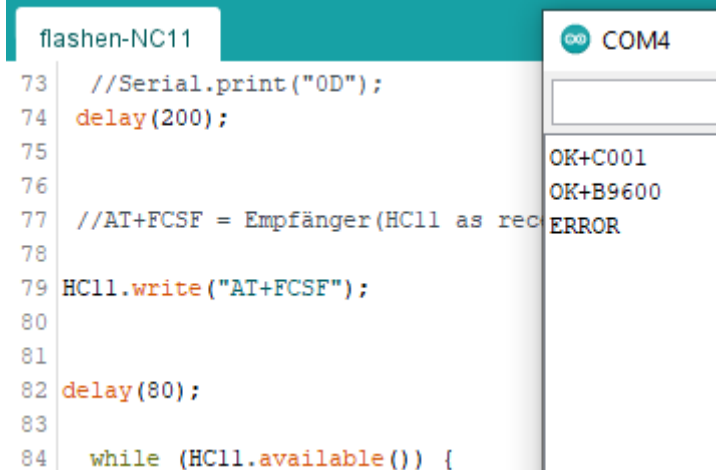
Ob nun aber der Baustein die Programmierung angenommen hatte, sollte sich jetzt zeigen.

Er sendet nicht! Es wird aber erkennbar Strom aufgenommen.

Zurück zur Konfiguration

Habe dann meine gut funktionierende Programmiersoftware geladen und aktiviert. Jetzt zeigte sich leider auch beim HC12 der ERROR!

```
flashen-NC11
73 //Serial.print("0D");
74 delay(200);
75
76
77 //AT+FCSF = Empfänger(HC11 as rec
78
79 HC11.write("AT+FCSF");
80
81
82 delay(80);
83
84 while (HC11.available()) {
```



Der letzte Befehl wurde nicht angenommen! Warum?

Jedenfalls ist jetzt dieser HC12 wieder umprogrammiert auf seine ursprüngliche Funktion. Er sollte nun auch wieder senden. Also machen wir den Test.

Erfolg: Drei arbeiten, davon einer sogar als HC11! Die zwei weiteren HC11 vermutlich sind kaputtgeflasht. Wo steht eigentlich geschrieben, wie oft man flashen darf?

Ich gebe auf. Es macht keinen Sinn, den o.g. Schaltungsvorschlag mit dem Client als Follower zu verwirklichen. Wenn es mir nicht gelingt, die AT+FCxx wirksam abzusetzen, dann wird so ein Paar auch nicht kommunizieren. Ich vermute, hier eine Baureihe eingekauft zu haben, die nicht dafür vorgesehen ist. Wahrscheinlich werden auch die moderneren HCxx nicht funktionieren, wenn man vom üblichen Mode abweicht.

Schade, denn die Schaltung erfüllt eigentlich meine Wünsche: Drei Kanäle zu steuern, die einen Motor für die Abstimmung einer Schleifenantenne treiben sollen. Aber da gibt es ja noch andere Wege. Ich werde es mit Schlüsseldaten im „Normalbetrieb“ versuchen. Allerdings bedingt das einen zweiten Arduino auf der Empfangsseite. Viel Aufwand, der mir nicht gefällt. Vielleicht nehme ich aber eine ganz normal Steckdosen-Fernsteuerung. Die sind einfach und preiswert. Ein besonderer Vorteil ist das fertige Gehäuse für die Fernbetätigung. Da muss ich nicht viel selbst basteln.

Wünsche allen Interessierten einen besseren Erfolg mit HCs. Vielleicht entdeckt jemand meine Fehler im Umgang mit den tollen Teilen.

DF8ZR; im Dez. 2020

es folgt umseitig meine Flashware:

```

/*
Programm : Empfänger Modul HC-11; bg
Funktion : Programm zum Setzen eines HC-11 als Empfänger/Sender
*/

#include <SoftwareSerial.h>

const int PIN_RX = 4; // connect with HC11 TX

const int PIN_TX = 5; // connect wirh HC11 RX

const int PIN_SET = 6; // connect with HC11 PIN5(SET)

SoftwareSerial HC11(PIN_RX, PIN_TX);

char SerialByteIn;           // Temporary variable
char HC11ByteIn;            // Temporary variable
String HC11ReadBuffer = ""; //Read/Write Buffer 1 for HC11/12
String SerialReadBuffer = ""; // Read/Write Buffer 2 for Serial

void setup()
{
pinMode(PIN_SET, OUTPUT);

digitalWrite(PIN_SET, HIGH); // HC11 nicht im commanmode
delay(80);                   // 80 ms delay before operation per datasheet

Serial.begin(9600);          // Open serial port to computer
HC11.begin(9600);           //startet HC11-PINS als serielle Schnittstelle

digitalWrite(PIN_SET, LOW); //schaltet HC-11 PIN5 set to AT mode(commandmode)
delay(80);

//Serial.println("AT+C015"); //set frequency to channel 15

HC11.write("AT+C001");
delay(80);

while (HC11.available()) {
  HC11ByteIn = HC11.read();

  Serial.print(HC11ByteIn);

}

```

```
delay(200);

//Serial.println("AT+B1200");           //1200 bd, slow for secure tranmission

HC11.write("AT+B9600");
delay(80);

while (HC11.available()) {
  HC11ByteIn = HC11.read();

  Serial.print(HC11ByteIn);
}

delay(200);

//AT+FCSF = Empfänger(HC11 as receiver), AT+FCMF = Sender(HC11 as transmitter)

HC11.write("AT+FCMF");

delay(80);

while (HC11.available()) {
  Serial.print(HC11ByteIn);
}
}

void loop()
{

//nur ein Durchlauf von Set-Up; leere void-loop

}
```