

# mor-sux, ein Morsetrainer nach der Koch- Methode

Ralf Beesner

3.6.2011

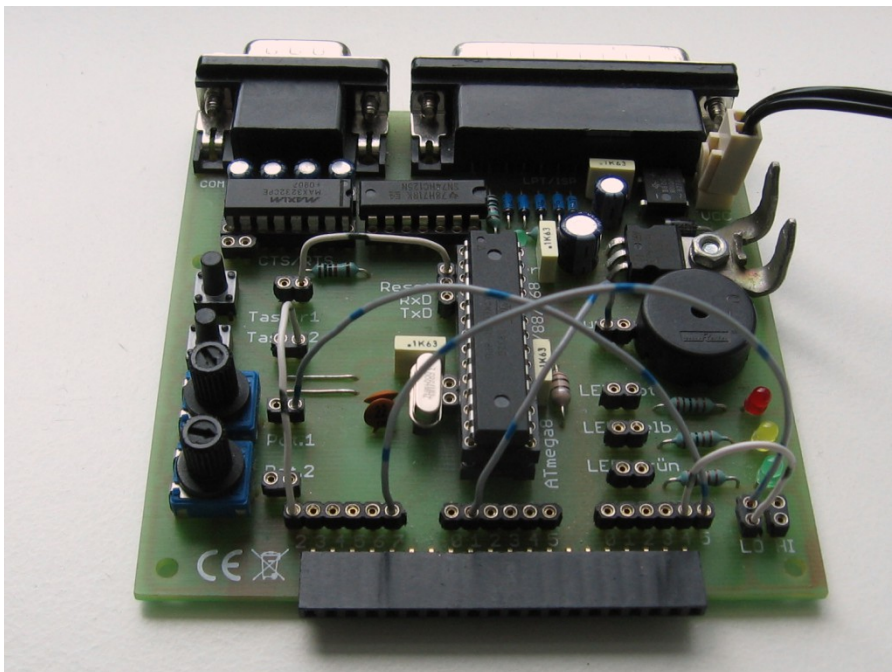


Abbildung 1: Versuchsaufbau

# 1 Einleitung

Roland Walter, DL7UNO, veröffentlichte auf seiner speziell eingerichteten Website [www.flohjagd.de](http://www.flohjagd.de) eine Bauanleitung für den Mini- Fuchsjagdsender „Foxy“, der neben einem ISM- Senderschaltkreis für 433,92 MHz mit einem ATtiny45 bestückt ist. Unter den Bascom- Beispielprogrammen findet sich eines, dass den Zufallszahlengenerator benutzt, um Telegrafie- Fünfer-Gruppen zu erzeugen.

Ich habe damit so lange herumgespielt, bis auf die Initialisierung des Zufallszahlengenerators nichts mehr übrigblieb; herausgekommen ist ein Morsetrainer mit zwei Modi:

Modus 1 erzeugt Fünfer- Gruppen mit wählbarem Zeichenumfang und wählbarer Geschwindigkeit; Modus 2 erzeugt Klartextwörter, die aus einem längeren Text zufällig zusammengewürfelt werden, auch hier ist die Geschwindigkeit einstellbar.

Diese „große Version“ habe ich lediglich auf einem Experimentiersystem für Atmega8 verdrahtet und getestet.

Auf einer Streifenrasterplatine realisiert habe ich aber eine kleinere Version mit einem Attiny24, die nur Modus 1, also Fünfergruppen „kann“.

## 2 Morsen lernen ?

Nur wenige Menschen werden heute noch Morsen lernen wollen; die Blütezeit des Amateurfunks ist eh vorbei und Morsetelegrafie wird in der Amateurfunkprüfung nicht mehr gefordert.

Aber „tot“ ist die Morsetelegrafie auf den Amateurfunkbändern noch lange nicht; sie ist reizvoller als Datenbetriebsarten wie PSK31, bei denen die Leute eigentlich nur noch ihre vorprogrammierten Standardtexte „abnudeln“, und Telegrafie benötigt weniger Sendeleistung und weniger gute Antennen als SSB- Sprechfunk.

Und morsen ist cool - chatten oder telefonieren kann ja jeder (im Internet) .

Als ich vor fast 40 Jahren Telegrafie im Selbststudium lernte, gab es nur Morsekurse auf Schallplatte oder Tonband. Die Länge der Lektionen war notorisch zu gering; irgendwann konnte ich Teile der Fünfer-Gruppen auswendig hinschreiben; bei Klartext reichte zwei- oder dreimaliges Abspielen, um große Teile aus dem Gedächtnis und nicht nach dem Höreindruck niederschreiben zu können.

Insbesondere das Üben mit Klartext ist jedoch wichtig, denn während man Fünfer-Gruppen irgendwann mechanisch ohne großes Nachdenken mitschreiben kann, kommt man bei der gleichen Geschwindigkeit bei Klartext „ins Schleudern“, weil man anfängt, die Buchstaben zusammenzusetzen und bestimmte Wörter zu erwarten. Wenn dann statt des erwarteten Wortes ein anderes erscheint, kommt man

völlig aus dem Takt.

Mit dem Aufkommen der Homecomputer und PCs wurde das Morsen lernen im Selbststudium stark vereinfacht; man war aber an das Gerät gebunden. Irgendwann brachte eine Schweizer Firma einen Morsetrainer auf Mikrocontrollerbasis namens „Morsix“ heraus, der es ermöglichte, ein kleines Kästchen mit sich herumzutragen und dann Morse zu üben, wenn man gerade etwas Zeit hatte (ohne auf Netzanschluss und PC angewiesen zu sein).

Das Gerät hatte seinen Preis und war sicherlich viel schöner als meine Lösung; aber meine ist billiger und selbstgemacht. Und der Name steht für was ganz anderes; er leitet sich von „more sucks“ ab ;-)

### 3 Hardware

Der Atmega8 arbeitet mit 1 MHz Takt aus dem internen RC- Oszillator; ein fabrikfrischer Atmega kann also ohne umfusen verwendet werden.

Die Stromversorgung erfolgt über eine 2-polige Klemme, an die z.B. ein Batteriekasten für 3 Mignonzellen angeschlossen werden kann.

Die Frequenz des Atmega8- 1- MHz- RC- Oszillators ist übrigens deutlich betriebsspannungsabhängig. Betreibt man den Mikrocontroller mit 3 V statt 4,5 bzw. 5 V, werden die Zeichen hörbar langsamer.

Über zwei „Mäuseklaviere“ mit 4 Bit bzw. 5 Bit werden Geschwindigkeit und Lektionsnummer vorgewählt. Gestartet wird das Programm über einen Resettaster. Abgeschaltet wird das Gerät über einen Taster an PD2/INT0, der den Prozessor in den Powerdown schickt, so dass er ständig an der Batterie verbleiben kann.

Die Lautstärke des Morsetons ist über das Poti R1 einstellbar, die Tonhöhe über das Poti R2.

Die Ausgabe der Morsetöne kann entweder über einen Buzzer oder eine Anschlussbuchse für Stereo- Ohrhörer erfolgen. Der Atmega8 erzeugt Rechtecksignale; der Buzzer dämpft durch seinen begrenzten Frequenzbereich und seine Kapazität die Oberwellen und die „Schaltknackse“ etwas.

Bei der Kopfhörer- Variante wird nicht nur der Pegel auf Kopfhörer- taugliche Werte herabgesetzt, sondern C2 und C3 bilden mit den Widerständen einen Bandpass, der die Schaltknackse und Oberwellen auf akzeptable Werte dämpft.

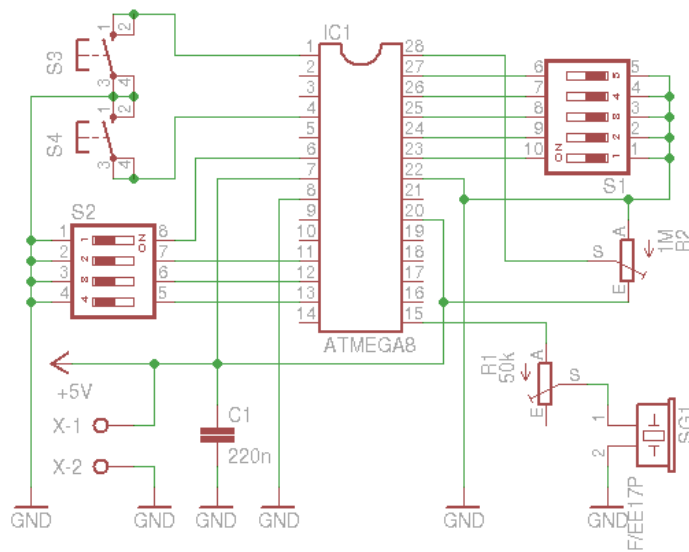


Abbildung 2: Version mit Buzzer

## 4 Software - Modus 1

Nach einem Druck auf den Reset- Taster fragt die Software die Stellung der „Mäuseklaviere“ für Geschwindigkeit und gewünschte Lektion ab.

Der Geschwindigkeitsbereich reicht von 15 Zeichen pro Minute (ZpM) bis 100 ZpM. Bei Geschwindigkeiten unter 70 ZpM werden die Pausen zwischen den Zeichen verlängert, während die Eigengeschwindigkeit der Zeichen weiterhin 70 ZpM beträgt. So lernt man auch bei geringen Geschwindigkeiten das charakteristische Klangbild der Zeichen (statt Punkte und Striche mitzuzählen).

Bei den Lektionsnummern 1-26 erfolgt die Ausgabe von Fünfer- Gruppen; Lektion 1 beginnt mit nur zwei Zeichen, mit jeder Lektion kommen zwei Zeichen dazu. Die Lektionen sind nach der Koch- Methode aufgebaut, die versucht, möglichst verschiedenartig klingende Zeichen hinzuzunehmen, damit man sich das Klangbild einprägt, statt Punkte und Striche abzuzählen.

Mit den drei letzten Lektionen kommen auch Umlaute dazu, die man sich nicht entgehen lassen sollte, wenn man im deutschsprachigen Raum herumfunken möchte.

Die auszugebenden Morsezeichen stecken in der Tabelle „Morsezeichentabelle\_1“. Die Reihenfolge der Zeichen entspricht der Koch- Methode. Der aus dem „Mäuseklavier“ ausgelesene Bytewert bestimmt die Obergrenze der Zahl, die der Zufallsgenerator „würfelt“.

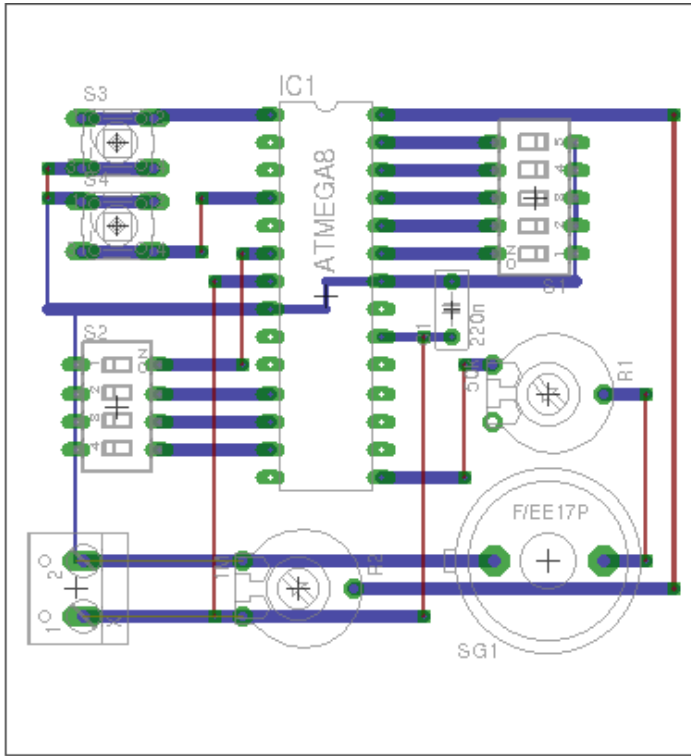


Abbildung 3: Version mit Buzzer

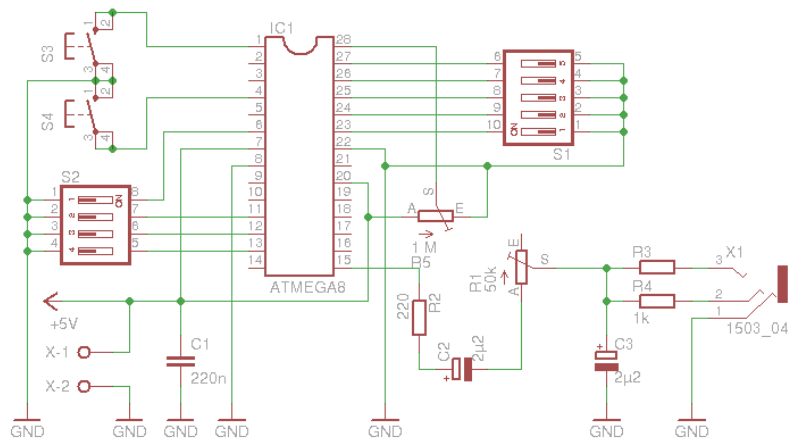


Abbildung 4: Version für Kopfhörer

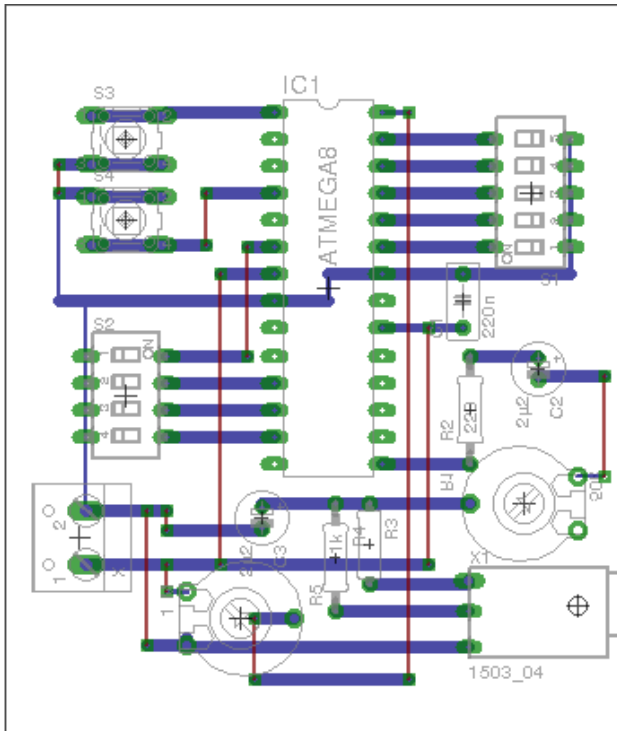


Abbildung 5: Version für Kopfhörer

## 5 Software - Modus 2

Lektionsnummern über 26 verzweigen in den Klartext- Modus; hier wird der komplette Zeichensatz zugrundegelegt und es werden aus einem längeren Text zufällig zusammengewürfelte Wörter wiedergegeben. Um es etwas schwieriger zu machen, werden auch Satz- und Sonderzeichen eingestreut.

Um Urheberrechtsproblemen aus dem Wege zu gehen, habe ich den zugrundeliegenden Text aus einem eigenen „Machwerk“ und aus zwei Beiträgen von Burkhard's Website zusammenkopiert,

Der Suchalgorithmus für ein Wort arbeitet in zwei Schritten: zunächst wird ein zufälliger Datastring der „Klartext\_tabelle“ gelesen: Im zweiten Schritt wird dann eine zufällige Position innerhalb des Strings angesteuert und ab dort nach einem Leerzeichen gesucht. Ab dem Leerzeichen wird das Wort in Telegrafie ausgegeben. Bei Erreichen eines weiteren Leerzeichens wird eine Wortpause eingeschoben und der Suchalgorithmus neu gestartet.

## 6 Software - Geschwindigkeitsmessung

Lektionsnummer 0 gibt das Wort „Paris“ aus. Es ist das Normwort, mit dem die Morsegeschwindigkeit gemessen wird. Die Wahl des Wortes „Paris“ wirkt heutzutage etwas skurril, aber Paris war (oder ist?) der Sitz der ITU, die mal die weltweiten Fernmelde- Standards festlegte, aber im Zeitalter des Internet und der der RFCs nach Bedeutung ringt.

## 7 eigene Texte

Die Datastrings in „Klartext\_tabelle“ kann man durch eigene Texte ersetzen. Voraussetzung ist der Besitz einer Vollversion von Bascom, da der Code 4 kByte überschreitet. Am besten eignen sich HTML- Texte oder Word- Files, da sie keine hart codierten Zeilenumbrüche aufweisen. Ein Data- String darf max. 254 Zeichen enthalten. Man muß daher den herangezogenen Text rechtzeitig umbrechen und eine neue Data- Zeile anlegen (ich habe die Grenze nicht bei 254 gewählt, sondern bei 155, da die Zeilen dann komplett auf meinen Monitor passten).

## 8 Flashen

Wer keinen Programmer für Atmega8 hat, kann ihn sich provisorisch aus 4 Bauteilen auf einem Steckbrett aufbauen und mit Burkhard Kainkas Programm ISPmega8.exe

flashen.

## 9 mor-sux und less-sux

Die Attiny24- Version habe ich „less-sux“ genannt. Wegen des kleineren Flash-Speichers mußte der Klartext- Modus wegfallen; less-sux „kann“ nur Fünfergruppen und den Paris- Testmodus.

Auf einen Attiny84 läßt sich die komplette mor-sux- Software mit minimalen Änderungen portieren, aber ich hatte nur einen Attiny24 zur Hand.

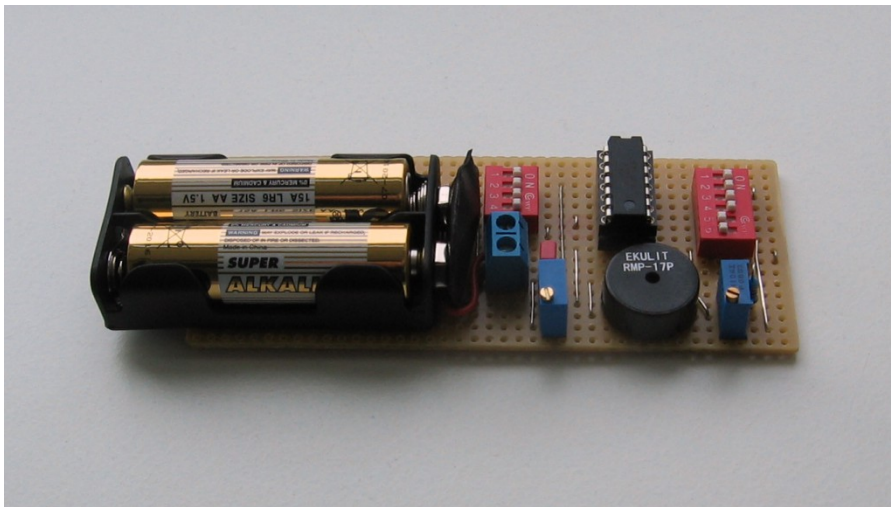


Abbildung 6: less-sux auf Streifenleiter-Platine

## 10 Hardware

Der ATtiny24 arbeitet ebenfalls mit 1 MHz Takt aus dem internen RC- Oszillator; ein fabrikfrischer Attiny24 kann also ohne umfusen verwendet werden.

Die Stromversorgung erfolgt über eine 2-polige Klemme, an die z.B. ein Batteriekasten für 2 Mignonzellen angeschlossen werden kann.

Die Taktfrequenz des Atiny24- RC- Oszillators ist nur wenig betriebsspannungsabhängig, daher wirkt es sich kaum aus, ob er mit 2 oder 3 Mignonzellen betrieben wird (lediglich die Lautstärke des Buzzers ist etwas geringer).

Die Bedienelemente entsprechen weitgehend der „mor-sux“- Version, allerdings gibt



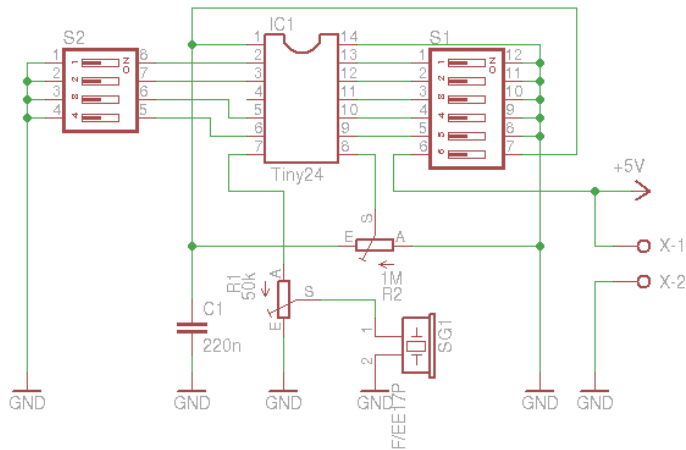


Abbildung 7: Schaltbild

es eine Ausnahme: da kein Port mehr frei war, ist das Ein- und Ausschalten herkömmlich über einen Schiebeschalter gelöst (Schalter 6 des Mäuseklaviers S1).

## 11 Bedienung

Zu Schluss die Bedienung in Kurzform:

- Wahl der Geschwindigkeit über das Mäuseklavier S2
  - Geschwindigkeiten: 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100 ZpM
- Wahl der Lektion über das Mäuseklavier S1
  - Lektion 0 : Paris- Schleife
  - Lektionen 1-26 : Fünfergruppen nach der Koch- Methode
  - Reihenfolge der Zeichen: K M U R E S N A P T L W I . J Z - F O Y , V G 5 / Q 9 2 H 3 8 B ? 4 7 C 1 D 6 Ø X = " ( ) + : ä ö ü ch KA
  - Lektion 27 : Klartext (entfällt bei less-sux)
- Start durch Drücken des Reset- Tasters S3 (mor-sux)
- Abschalten durch Drücken des Tasters S4 (mor-sux)
- Ein- Ausschalten und Reset durch Schalter 6 des Mäuseklaviers 1 (less-sux)

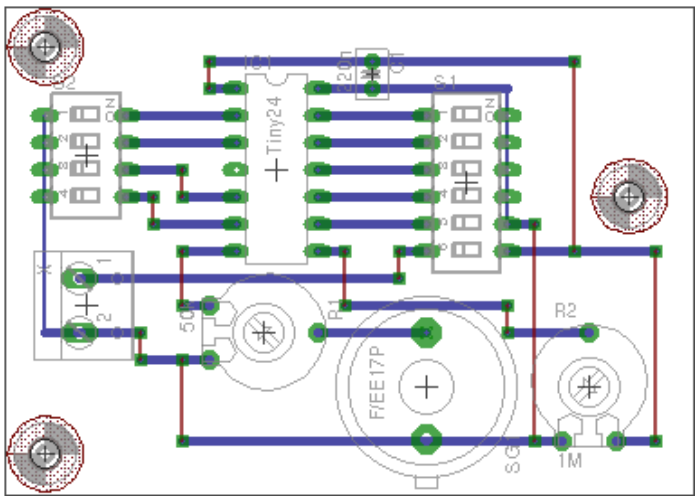


Abbildung 8: Aufbauvorschlag auf Streifenleitungsplatine