

Handbuch

metaMorserino

Ein multifunktionales Morsegerät

Version 3.x



Inhaltsverzeichnis

| | |
|--|----|
| Was ist metaMorserino? | 3 |
| Was ist das Besondere am metaMorserino? | 3 |
| Geräte-Anschlüsse..... | 4 |
| Bedienvorrichtungen:..... | 5 |
| Einschalten, Auswahl des Betriebsmodus..... | 5 |
| Bedienung im Modus CW Keyer..... | 7 |
| Einstellen der Parameter..... | 7 |
| Speicherung der Parameter..... | 9 |
| Zu den Keyermodi..... | 9 |
| Zu den Einschränkungen eines Touch Keyers..... | 10 |
| Bedienung im Modus CW Trainer..... | 11 |
| Bedienung im Modus CW Decoder..... | 14 |
| Wie funktioniert der CW Decoder?..... | 14 |
| Anhang: Kalibrierung des Touch Paddles..... | 15 |
| Warum kalibrieren?..... | 15 |
| Anhang: Die generierten bzw. dekodierten Morsezeichen..... | 16 |

Was ist metaMorserino?

MetaMorserino ist ein kleines Gerät zum Eingeben und zum Trainieren von Morsezeichen. Es kommt als Bausatz, der leicht und schnell aufzubauen ist, besteht aus extrem wenigen Teilen (weil möglichst viel durch Software erledigt wird), und verfügt dabei über einen erstaunlichen Funktionsumfang:

- CW Keyer mit Touchpaddles (das ist die zeitgemäße Art von Morsetaste, die Punkte und Striche automatisch erzeugt, wenn man einen der beiden Sensoren berührt; CW ist Amateurfunkjargon für Morsen, „Keyer“ ist englisch für Morsegeber, Paddles nennt man Morsetasten mit normalerweise zwei vertikal stehenden „Tasten“, und „Touchpaddles“, weil es sich nicht um mechanische „Schalter“ handelt, sondern um Berührungssensoren).
- CW Trainer (Morseübungsgenerator, der Zeichen in zufälliger Reihenfolge ausgibt, damit man das „Mitlesen“ von Morsezeichen trainieren kann - „Lesen“ tut man übrigens Morsezeichen normalerweise mit dem Gehör, und nicht mit den Augen!).

Durch ein späteres „Upgrade“, das aus ein paar Bauteilen bestehen wird, die noch auf der Platine Platz finden, kann noch eine dritte Funktion integriert werden, die softwareseitig schon integriert ist:

- • CW Decoder (also ein Gerät, das man an einen Funkempfänger anschließt – üblicherweise an den Kopfhörerausgang –, und welches dann die mit dem Empfänger als Töne empfangenen Morsezeichen dekodiert und auf dem Display anzeigt). Es kann auch direkt eine Handtaste angeschlossen werden, und die damit eingegebenen Morsezeichen werden dekodiert.

Das „Herzstück“ des Gerätes ist ein Arduino Nano, zu dem als wesentliche Komponenten ein zweizeiliges LCD Display, ein Drehgeber mit Drucktaster und ein Lautsprecher hinzugefügt sind. Der Name metaMorserino ist eine Wortkombination aus „MetaFunk“, „Morsen“ und „Arduino“.

Was ist das Besondere am metaMorserino?

Minimalistische Hardware, damit extrem kostengünstig, und möglichst viel der Funktionalität in die Software verlagert:

- einfach aufzubauender Bausatz mit wenigen Teilen
- keine eigene Sensor-Elektronik für die Touch Paddles, analoge Arduino Eingänge werden für diese Funktionalität verwendet
- Lautstärkeregelung per Software durch niederfrequente Pulsbreitenmodulierung eines hochfrequenten pulsbreite-modulierten Trägersignals (ein etwas unsauberer Tonsignal mit fallweisen Klicks wird dabei in Kauf genommen)
- mehrere Funktionen in einem Gerät (CW Geber, CW Übungsgerät,...)
- Erweiterungsfähigkeit hardware- und softwareseitig (Erweiterung zu einem Morsedekoder)

Geräte-Anschlüsse

Kopfhörer

für handelsübliche Stereo Kopfhörer

Getasteter Ausgang zum Funkgerät (RCA Buchse, Cinch Buchse)

Damit kann ein Sender getastet werden (das Betreiben eines Senders ist lizenzierten Funkamateuren vorbehalten!)

Achtung, der Keyer darf nur an Funkgeräte angeschlossen werden, die eine **positive** Spannung von **max. 30V** gegen Ground tasten - das sind die allermeisten modernen Transceiver, aber bitte **vorher überprüfen**! Wird eine negative (oder höhere) Spannung getastet, muss ein Relais verwendet werden!

USB Mini Anschluss

kann zum Neuprogrammieren („Flashen“) des Arduino verwendet werden, oder auch zur **Stromversorgung** über ein USB Netzgerät (5V).

Im Gehäuseinneren: 9V Batterie

Zink-Kohle Batterien sind nicht zu empfehlen, die sind sehr schnell leer. Am besten gute Alkali-Mangan Batterien, oder wiederaufladbare Akkus in derselben Bauform. Lithium Batterien haben die größte Kapazität, sind allerdings empfindlich teuer!

Im Gehäuseinneren: Experimentierplatine

Auf der Platine gibt es einen Bereich, der als Lochrasterfeld ausgeführt ist; dort können weitere Bauteile untergebracht und verlötet werden, zB eine Verstärkerschaltung, damit man Tonsignale dekodieren kann. Auch sind alle noch unbenutzten Pins des Arduino dort herausgeführt, so dass das Gerät eigentlich eine sehr universelle Arduino-Experimentierplattform darstellt.

Bedienvorrichtungen:

Ein/Aus Schalter

Dieser schaltet nur die 9V Batterie, wird über USB versorgt, ist immer eingeschaltet, und der Schalter sollte auf Stellung AUS bleiben!

Drehgeber m. Drucktaster

Dient zur Einstellung der Gebegeschwindigkeit und zum Einstellen diverser Konfigurationsparameter.

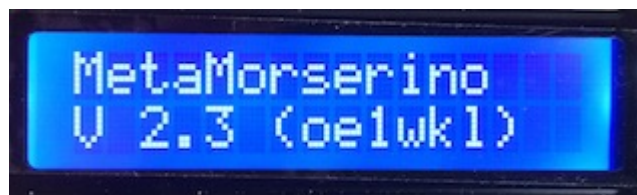
Sensorpaddle

Es handelt sich um Platinen mit Kupferflächen, durch Lötstoplack geschützt. Damit werden durch Berühren auf der einen Seite Punkte ausgegeben, und auf der anderen Seite Striche.

Einschalten, Auswahl des Betriebsmodus

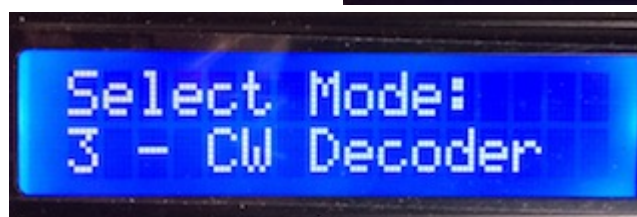
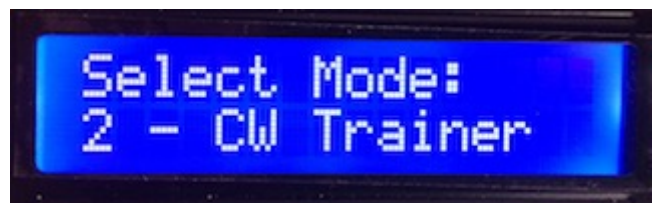
Das Gerät wird entweder durch Anstecken an eine USB Stromversorgung mittels eines Mini-USB Kabels eingeschaltet werden (USB Steckernetzteil, z.B. Handy Ladegerät, oder PC etc.), oder – wenn im Gehäuse eine 9V Blockbatterie angeschlossen wurde – durch Betätigen des Ein/Aus-Schalters.

Das Display leuchtet, und es erscheint für kurze Zeit ein Begrüßungsbildschirm.



Anschließend befindet man sich im Hauptmenü, welches eine Auswahl des Betriebsmodus ermöglicht. Derzeit sind die Modi „CW Keyer“ und „CW Trainer“ implementiert, ein Nachrüsten für einen dritten Modus („CW Decoder“) ist vorgesehen.

Im Modus CW Keyer erlaubt das Gerät, über die beiden Touch Paddles Morsezeichen zu generieren, im Modus CW Trainer werden zufällige Morsezeichen generiert, um so eine Übungsmöglichkeit für diejenigen zu schaffen, die ihre Morsekenntnisse und vor allem ihre Geschwindigkeit im Aufnehmen von Morsezeichen verbessern wollen.



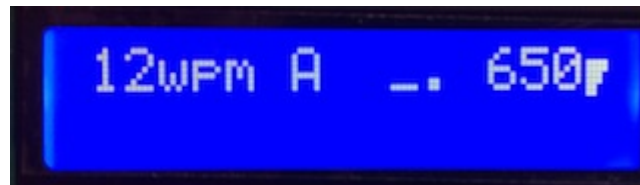
Der Modus wird durch Drehen des Drehgebers und kurzen Druck auf den Knopf des Drehgebers (Drucktaster) gewählt. Es wird kurz angezeigt, welcher Modus gestartet wird, und dann erscheint das für den betreffenden Modus normale Bild am Display.

Je nach dem gewählten Modus können diverse Parameter eingestellt werden.

Bedienung im Modus CW Keyer

Durch Betätigen der beiden Sensoren werden Punkte und Striche ausgegeben, die in ihrer Kombination dann die gewünschten Morsezeichen ergeben sollen.

Werden „gültige“ Morsezeichen erzeugt, so werden diese in der unteren Zeile des Displays angezeigt, wobei die Zeichen von rechts nach links durch das Display laufen. Außerdem kann über die Cinch-Buchse (RCA Buchse) ein Funkgerät angeschlossen werden, die generierten Punkte und Striche werden dann auch über das Funkgerät gesendet.



Es werden in der oberen Zeile des Displays die eingestellten Parameter angezeigt, in folgender Reihenfolge:

- Gebegeschwindigkeit (in Wörtern pro Minute - wpm)
- Keyermodus (siehe unten)
- Polarität der Paddles (angezeigt als _ oder . je nachdem ob die Punkte links oder rechts gegeben werden)
- Tonhöhe in Hz
- Lautstärke als höher und breiter werdender Balken

Durch **Drehen des Drehgebers** kann die **Gebegeschwindigkeit** eingestellt werden, von 5 - 30 wpm (1 wpm entspricht 5 Buchstaben pro Minute, das Referenzwort für die Definition der Geschwindigkeit ist PARIS, dh. bei eingestelltem Tempo 12 wpm kann man 12 mal pro Minute das Wort PARIS geben, bzw. die darin enthaltenen 60 Buchstaben und Wortabstände).

Durch kurzes **Betätigen des Tasters** am Drehgeber gelangt man in das **Konfigurationsmenü** des CW Keyers, und es können die diversen Parameter eingestellt werden. Im Konfigurationsmenü ist lediglich der gerade einzustellende Parameter in der oberen Zeile zu sehen, und ganz links zwei Pfeilchen, die andeuten, dass der betreffende Parameter durch Drehen am Drehgeber geändert werden kann.

Durch langes Gedrückthalten im Normalbetrieb gelangt man ins Hauptmenü, wo man gegebenenfalls einen anderen Modus auswählen kann.

Einstellen der Parameter

Durch kurzen Klick auf den Knopf des Drehgebers kommt man von einem Parameter zum nächsten, angezeigt durch den Unterstreichungsstrich, bzw. durch Doppelklick zurück zum vorigen Parameter.

Ein langer Klick (Knopf mindestens eine halbe Sekunde gedrückt halten) führt in jedem Fall zurück zum Normalbetrieb. Ist der gewählte Parameter die Lautstärke, gelangt man auch mit einem kurzen Klick in den Normalbetrieb, und ist der Parameter der Keyermodus, dann führt ein Doppelklick zum Normalbetrieb.

(Oder anders ausgedrückt: ein Klick führt immer nach rechts um eine Position weiter in den Einstellungen, ein Doppelklick nach links, und ein langer Klick führt immer an die erste Position, und das ist der Normalbetrieb. Ist man schon im Normalbetrieb, gelangt man durch den langen Klick ins Hauptmenü.)

Durch Drehen des Drehgebers können folgende Parameter gewählt werden:

Keyermods

Curtis A, Curtis B, verbesserter Curtis B (B+) oder Ultimatic können gewählt werden.+, angezeigt durch **A**, **B**, **B+** und **Ul**. Siehe dazu weiter unten.



Polarität

Da die persönlichen Vorlieben unterschiedlich sind, können die Punkte wahlweise auf der linken oder auf der rechten Seite des Paddles eingegeben werden. Am Display wird gezeigt was links bzw. rechts liegt.



Tonhöhe

Von 250 Hz bis 950 Hz in Schritten von 50 Hz einstellbar.



Lautstärke

Diese ist in 8 Schritten von lautlos bis maximaler Lautstärke einstellbar und wird durch einen höher und breiter werdenden Balken rechts im Display angezeigt.



Speicherung der Parameter

Beim Verlassen der Konfiguration bzw. bei der Rückkehr in den Normalbetrieb werden die eingestellten Werte (inklusive der gerade eingestellten Geschwindigkeit) im EEPROM des Arduino gespeichert, und beim Einschalten wieder eingestellt - damit bleiben die Konfigurationsparameter auch beim Ausschalten gespeichert und müssen nicht jedes Mal neu eingegeben werden.

Eine Änderung der Geschwindigkeit (im Normalbetrieb) wird erst gespeichert, sobald mindestens 12 Zeichen in dieser Geschwindigkeit eingegeben wurden.

Zu den Keyermodi

Aufgrund der technischen Entwicklungen haben sich unterschiedliche Verhaltensweisen bei elektronischen Keyern heraus kristallisiert. Die Unterschiede liegenender allem im Verhalten, wenn beide Paddles gleichzeitig gedrückt (in unserem Fall: berührt) werden (im Englischen spricht man dann von "squeeze").

In jedem Fall beginnt der Keyer mit jenem Element (Punkt oder Strich), das zuerst gedrückt (berührt) wurde.

Man unterscheidet einerseits zwischen "**lambic Mode**"¹ und "**Ultimatic Mode**"² - im lambic Mode werden Punkte und Striche abwechselnd gegeben, solange beide Paddles bedient werden, im Ultimatic Mode wird nach einem einmaligen ersten Element das andere Element wiederholt ausgegeben, solange beide Paddles gedrückt sind.

Also lambic: _ . _ . _ . usw. bzw. . _ . _ . _ usw.,

aber Ultimatic : _ usw. bzw. . _ _ _ _ usw.

Beim lambic Mode haben sich noch weitere Unterschiede herausgebildet, die letztlich mit dem Timing zusammenhängen, wann die Paddles wieder losgelassen werden.

Im **lambic Mode B** wird bei Loslassen während des letzten Elements noch ein gegensätzliches "nachgeschoben" (also wenn das letzte Element ein Strich war, und man lässt während des Strichs beide Paddles aus, wird noch ein Punkt hinzugefügt - irgendwie sehr gewöhnungsbedürftig³. Man muss also das Punkt-Paddle schon auslassen, bevor noch der Strich gegeben wird, wenn man nicht diesen nachgeschobenen Punkt will.

Im **lambic Mode A** gibt es dieses nachgeschobene Element nicht, die Bedienung ist daher ein wenig intuitiver. Die meisten US-amerikanischen Hersteller verwenden üblicherweise Mode A, die meisten japanischen Geräte benutzen Mode B (zB der beliebte FT817!).

Man hat später auch noch einen "erweiterten" bzw. "verbesserten" Mode B entwickelt (den ich **B+** nenne) - da hängt es davon ab, wann man genau das zweite Paddle los lässt - in unserem Beispiel, bei dem das letzte Element ein Strich war, wird im Mode B immer ein Punkt nachgeschoben, wenn man erst während des Strichs das Punkt-Paddle loslässt; im Mode B+ kann man das Paddle auch noch etwa im ersten Drittel des Strichs loslassen, ohne dass ein Punkt nachgeschoben wird - erst wenn das Loslassen später erfolgt, kommt der Punkt noch nach.

Generell muss man sagen, dass man für jeden Modus einige Zeit üben muss, bevor man ihn wirklich fehlerfrei beherrscht. Was einem besser liegt ist natürlich Geschmacksache - man muss aber bedenken, dass in Funkgeräten eingebaute Keyer oft nur einen bestimmten Modus unterstützen - das

¹ Die Bezeichnung kommt vom lat. Iambus, das ist ein Versfuß, bei dem einem Breve (kurz/leicht) ein Longum (lang/schwer) folgt, wie bei beim Morsezeichen für „a“ (kurz – lang). Natürlich hätte man den Modus auch „Trochaic“ nennen können, das ist der Versfuß lang – kurz. Siehe <https://de.wikipedia.org/wiki/Iambus>

² Kunstwort aus lat. Ultimatus und *automatic*, weil das zuletzt eingegebene Element automatisch wiederholt wird.

³ Angeblich rührt dieses Verhalten eigentlich von einem Programmierfehler her, der der Fa. Curtis unterlaufen war, die als erste einen Chip entwickelte, der den lambic Mode implementierte.

spricht auf jeden Fall dafür, einen externen Keyer zu verwenden, bei dem man den Modus einstellen kann.

Selbstverständlich kann man einen Keyer auch so benutzen, dass man gar nicht alle beiden Paddles gemeinsam bedient - dann verschwinden natürlich auch die Unterschiede in den Keyer Modi (das ist das Äquivalent eines mechanischen Single-Lever Paddles, welches nur über einen Hebel verfügt, der nach links oder rechts bewegt werden kann).

Warum benutzt man überhaupt die Squeeze Technik? Nun, man kann etwas an Bewegung sparen, und damit die Morsezeichen etwas effizienter geben, was vor allem bei höheren Geschwindigkeiten etwas ausmacht. Man darf diesen Effizienzgewinn allerdings auch nicht überbewerten:

Während der Übergang von der normalen Handtaste zum Keyer (ohne Squeeze) ein ziemlich großer Sprung ist (laut K7QO braucht man, um die Buchstaben von A - Z und die Ziffern von 0-9 einmal zu geben, mit der Handtaste 132 Handbewegungen, und mit einem Keyer ohne Squeeze nur 73, das sind fast 45% weniger!), gewinnt man durch die Squeeze Technik nur mehr relativ wenig: Man braucht dann 65 Handbewegungen - also etwa 50% weniger als bei der Handtaste, wobei es ziemlich egal ist, welchen Modus man verwendet - dieselbe Einsparung gilt auch für den Ultimatic Modus.

Das ist allerdings eine Milchmädchenrechnung, denn die einzelnen Zeichen sind in normal vorkommendem Text nicht gleichmäßig verteilt, die häufigeren Zeichen sind im Morsealphabet auch meist die ohnedies kürzeren; letztlich heißt das, dass die Squeeze-Technik wirklich nur einen marginalen Vorteil gegenüber dem Keyer ohne Squeeze bringt. Ob man viel Zeit dafür aufwenden soll, sich eine Squeeze-Technik anzueignen, muss natürlich jeder für sich entscheiden....

Mehr Infos zu den diversen Meyer Modi findet man unter diesen Adressen im Internet:

<http://www.qth.net/pipermail/agcw/2003-July/000385.html>

http://www.dl2lto.de/sc/TM_tab_iambic.htm

<http://wb9kzy.com/ultimat.txt>

<http://www.gsl.net/dk5ke/squeezen.html>

http://www.qrpproject.de/Media/pdf/IambicModeA_B.pdf

<http://www.morsex.com/pubs/iambicmyth.pdf>

https://en.wikipedia.org/wiki/Telegraph_key

<http://www.palm-radio.de/pdf/IambicPaddleModeAorBfunction-1.pdf>

Zu den Einschränkungen eines Touch Keyers

Wirklich schnell geben kann man nur mit einem hochwertigen mechanischen Paddle, das dann auch dementsprechend teuer ist. Dies liegt daran, dass die Zeitspanne, die ein Paddle gedrückt sein darf, bei hohen Geschwindigkeiten extrem kurz ist (ein Punkt dauert bei Tempo 30 wpm - 150 BpM nur 40 msec!)

Mit Sensoren jeder Art (ob kapazitiv wie bei unserem Keyer, oder ob resistiv) ist es sehr schwierig, zuverlässige "Schaltzeiten" unter etwa 75 msec zu erhalten. Das heißt aber, dass es sehr schwierig ist, bei einem Tempo größer als etwa 15 wpm (75 Buchstaben pro Minute) Fehleingaben zu vermeiden (man erhält leicht einen Strich oder Punkt mehr als eigentlich intendiert). Besonders gravierend wirkt sich das bei den Curtis-Modi B und B+ aus, da da schon während der Dauer des ausgegebenen Elements (Punkt oder Strich) schon das Paddle auf weitere Eingaben geprüft wird.

Will man das Tempo mit dem Touch Keyer etwas forcieren, sind daher die Modi Curtis A oder Ultimatic die bessere Wahl.

Bedienung im Modus CW Trainer

In diesem Modus gibt das Gerät Morsezeichen in zufälliger Reihenfolge aus, und man kann so seine Fertigkeit im Aufnehmen von Morsezeichen üben.

Die Ausgabe der Morsezeichen beginnt, wenn man beide Paddles gleichzeitig betätigt („Squeeze“). Den Squeeze benutzt man auch, um die Ausgabe der Morsezeichen wieder zu **beenden**.

Die generierten Zeichen werden als Töne ausgegeben (Lautsprecher oder Kopfhörer) und zusätzlich, zur Kontrolle, in der unteren Zeile des Displays angezeigt.

Auch im Übungsgeneratormodus werden in der oberen Zeile die gewählten Parameter angezeigt und können in ähnlicher Weise wie im CW Keyer Modus geändert werden:



Geschwindigkeit (in wpm)

Dies entspricht auch der Geschwindigkeit der Morsezeichen im CW Keyer Modus.

Durch **Klicken** kommt man, wie beim CW Keyer Modus, zu den diversen **Konfigurationsmöglichkeiten**:

Farnsworth-Abstands-Faktor

Angezeigt durch ein **Sternchen** und eine **Zahl** zwischen 1 und 9. Diese Zahl gibt den Faktor an, um welchen die Pausen zwischen den Zeichen und zwischen den Wörtern gedehnt sind, um so mehr Zeit zum Erfassen (und ggf. Aufschreiben) der Morsezeichen zu erhalten - man nennt dies auch die "Farnsworth Methode".

Normalerweise sind folgende Pausen bzw. Abstände üblich:

- zwischen den Elementen eines Zeichens (Punkten und Strichen): so lange wie ein Punkt
- zwischen den Buchstaben: so lange wie drei Punkte, bzw. wie ein Strich
- zwischen Wörtern: so lange wie 7 Punkte, bzw. etwas länger als zwei Striche

Bei der Farnsworth-Methode werden die Abstände zwischen den Buchstaben und Wörtern verlängert. Man lernt die Zeichen gleich mit hoher Geschwindigkeit (um zu vermeiden, dass man Punkte oder Striche mitzählt - man soll den Rhythmus des Zeichens im Ohr haben), kommt aber dem Lernenden durch größere Abstände entgegen (dh. die tatsächlich übertragenen Buchstaben oder Wörter pro Minute sind deutlich weniger).



Bei unserem Gerät wird nach dem Sternchen der **Faktor** für die Verlängerung der Abstände angezeigt, zwischen 1 (volles Tempo, keine längeren Abstände) bis 9 (9 mal so lange Abstände).

Empfohlen wird, immer mit mindestens Tempo 18-20 wpm zu üben, und den Verlängerungsfaktor

beginnend bei 8 oder 9 immer weiter herabzusetzen; kommt man mit Faktor 4 gut zurecht, setzt man das Tempo noch etwas hinauf (21-24 wpm), und verlängert wieder die Abstände. Dies so lange, bis man auch die hohen Tempi gut beherrscht.

Wichtig ist, dass die Pausen nicht zu lange sind – man sollte immer „ein bisschen Stress“ haben beim Aufnehmen der Zeichen, wenn man sich auf höheres Tempo trainieren will.

Anmerkung:

Eine genaue Beschreibung der Farnworth Methode gibt es auf der Website der ARRL:

<http://www.arrl.org/files/file/Technology/x9004008.pdf>

Auch die Koch Methode beruht u.a. darauf, von Anfang an die Zeichen in möglichst hohem Tempo zu geben. Zur Koch-Methode siehe <http://www.df2ok.privat.t-online.de/afu01a.htm#KOCHLINKS>

Beim Standardwort für die Berechnung der Übertragungsgeschwindigkeit für Morsezeichen ("PARIS") benötigen wir 50 Zeiteinheiten in der Länge jeweils eines Punktes; davon entfallen 19 Zeiteinheiten tatsächlich auf die Punkte und Striche und die Abstände zwischen Punkte und Strichen, und 31 Zeiteinheiten auf die Abstände zwischen Buchstaben (und genau einem Wortabstand).

Verlängern wir die Abstände zwischen Buchstaben und Wörtern auf das 9-fache werden aus den 31 Zeiteinheiten 279, macht mit den 19 Zeiteinheiten für die Zeichen selber 298 Zeiteinheiten. Wir haben also die benötigte Zeit für die Übertragung auf etwa das 6-fache gedehnt.

Bei einem Verlängerungsfaktor von 9 wird also ein Text effektiv nur mit etwa 1/6 der eingestellten Zeichengeschwindigkeit übertragen:

Eingestellt 18 wpm (90 BpM) - effektive Geschwindigkeit ca 3 wpm (15 BpM)

Bei Verlängerungsfaktor 6 ist die effektive Übertragungsrate ca. 1/4:

Eingestellt 20 wpm (100 BpM) - effektive Geschwindigkeit ca 5 wpm (25 BpM)

Und bei Verlängerung um den Faktor 3 haben wir in etwa die halbe Übertragungsrate:

Eingestellt 24 wpm (120 BpM) - effektive Geschwindigkeit ca 12 wpm (60 BpM)

Bei Verlängerung der Pausen auf das Doppelte liegen wir bei einer Übertragungsrate von etwa 60%:

Eingestellt 20 wpm (100 BpM) - effektive Geschwindigkeit ca 12 wpm (60 BpM)

Ausgabe der Morsezeichen

Es werden entweder 5er Gruppen ausgegeben, dh. immer 5 Zeichen gefolgt von einer Pause in der Länge eines Wortabstands, oder Zeichenfolgen, die die Struktur von Amateurfunk-Rufzeichen haben. Bevor mit der Ausgabe der 5er Gruppe oder der Rufzeichen begonnen wird, wird zuerst immer die Sequenz „**v v v <ka>**“ ausgegeben, damit man Zeit hat, sich auf die eigentliche Ausgabe vorzubereiten. <ka> (-.-) ist eine sogenannte „Betriebsabkürzung“ (engl. „Pro Sign“) und bedeutet „Beginn der Übertragung“.

Textauswahl

Im mittleren Feld der oberen Displayzeile wird angezeigt, was als "Übungstext" ausgegeben wird. Grundsätzlich gibt es die Wahl, verschiedenen Arten von 5er Gruppen ausgeben zu lassen, oder Zeichenfolgen, die die Struktur von Amateurfunk-Rufzeichen haben.

a : 5-er Gruppen von Buchstaben a-z

9 : 5-er Gruppen von Ziffern 0-9

- a9 : 5-er Gruppen von Buchstaben und Ziffern (a-z, 0-9)
a9? : 5-er Gruppen von Buchstaben, Ziffern und Interpunktionszeichen
?> : 5-er Gruppen von Interpunktionszeichen und Betriebsabkürzungen
a9?> : 5-er Gruppen von Buchstaben, Ziffern, Interpunktionszeichen und Betriebsabkürzungen
ää : 5-er Gruppen von Buchstaben inklusive deutsche Sonderzeichen (ä,ö,ü,ch)
ää9 : dasselbe, zusammen mit Ziffern
ää9?> : dasselbe, mit Ziffern, Interpunktionen und Betriebsabkürzungen

CALLS: Rufzeichen. Diese haben das Format von Amateurfunk-Rufzeichen, es sind aber zufällig generierte Zeichen, so dass auch Rufzeichen ausgegeben werden, die in der tatsächlichen Praxis so nicht vorkommen. Manche der Rufzeichen sind auch mit einem nachfolgenden /m oder /p versehen, da diese Erweiterungen auch relativ häufig sind.



(siehe den Anhang zu den verwendeten Morsezeichen)

Tonhöhe des Übungsgenerators (wie beim CW Keyer)

Lautstärke des Übungsgenerators (wie beim CW Keyer)

Während im Keyermodus über den Tastenausgang (RCA Buchse, Cinchbuchse) ein Sender getastet werden kann, wird dies im Übungsmodus (aus verständlichen Gründen) nicht gemacht, dh. es wird nur über Lautsprecher bzw. Kopfhörer ausgegeben.

Allerdings wird, wie beim Keyer, in der unteren Zeile des Displays auch angezeigt, welche Zeichen gerade generiert wurden (dh. am besten Hören und im Kopf Dekodieren, dann kurzer Blick aufs Display zur Kontrolle)!

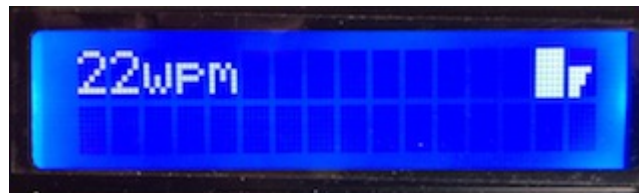
Bedienung im Modus CW Decoder

In diesem Modus gibt es keine weiteren Konfigurationsmöglichkeiten. Durch Druck auf den Knopf des Drehgebers (lang oder kurz ist hier egal) kommt man wieder zurück in das Hauptmenü.

Das Drehen des Drehgebers verändert in diesem Modus die Lautstärke des ausgegeben Tonsignals.

Wie funktioniert der CW Decoder?

Um den CW Decoder benutzen zu können, muss man entweder eine Morsetaste an den Pin D12 des Arduino anschließen (der zweite Anschluss der Taste geht nach Masse/Ground), oder über eine einfache Verstärkerschaltung (alle Teile finden noch Platz auf der Platine) ein Tonsignal an den Arduino-Pin A6. Dieses Tonsignal kann zB. Von einem geeigneten Kurzwellenempfänger kommen, mit dem man Morsezeichen empfangen kann. In letzterem Fall versucht metaMorserino Tonsignale in der Höhe von 744 Hz zu erkennen. Die Tonsignale müssen allerdings halbwegs störungsfrei und rauschfrei sein!



Wird die Morsetaste gedrückt oder ein 744 Hz Ton erkannt, wird ein ebensolcher Ton am Lautsprecher ausgegeben, und es erscheint auch ein blockartiges Symbol neben dem Lautstärke-Symbol am Display. Die Software versucht dann, die so eingegebenen Morsezeichen zu dekodieren, und stellt die dekodierten Zeichen dann als Laufschrift in der unteren Zeile des Displays dar. Dabei passt sich die Software ständig an das Tempo der erkannten Morsezeichen an – bei sehr abruptem Tempowechsel kann es aber ein paar Zeichen dauern, bis wieder korrekt dekodiert wird.

Die aktuelle Geschwindigkeit der dekodierten Morsezeichen wird links in der oberen Displayzeile angezeigt.

Anhang: Kalibrierung des Touch Paddles

Werden beim Einschalten beide Paddles betätigt (sog. "Squeeze"), geht metaMorserino in den Kalibrierungsmodus. Während bei jedem Einschalten der Ruhezustand der Paddles kalibriert wird, wird im Kalibriermodus auch die Sensitivität im betätigten Zustand kalibriert, und die ermittelten Werte werden im EEPROM gespeichert.

Das Gerät meldet sich mit "Calibration" am Display, kurz darauf wird man aufgefordert "Release Paddles!", also beide Paddles loslassen. Die ermittelten Werte werden kurz am Display angezeigt (diese Werte haben aber für den Benutzer keinerlei Bedeutung).

Normalerweise ist keine manuelle Kalibrierung notwendig, da die Software ständig während der Eingabe eine automatische Rekalibrierung vornimmt!

Warum kalibrieren?

Die Sensitivität der Paddles hängt von diversen äußeren Einflüssen ab, unter anderem auch davon, ob weitere Geräte am metaMorserino angeschlossen sind (ein PC, ein USB Ladegerät, ein Funkgerät, Kopfhörer etc.). Während sich die gemessenen Sensorwerte im Ruhezustand nur wenig unterscheiden, kann es bei Berührung zu relativ starken Abweichungen kommen. Ist die Sensitivität nicht korrekt kalibriert, kann es vorkommen, dass entweder ein Berühren nicht zuverlässig erkannt wird, oder der Sensor überempfindlich ist und schon bei Annäherung (ohne Berührung) aktiviert wird.

Falls einer der geschilderten Effekte auftritt, und sich das Verhalten nicht durch die Eingabe von ein paar Zeichen bessert, empfiehlt es sich, das Gerät auszuschalten und beim Einschalten manuell zu kalibrieren, wie oben beschrieben.

Anhang: Die generierten bzw. dekodierten Morsezeichen

| Buchstabe | Morse Code | Wird dekodiert | Beim Trainer enthalten in |
|-----------|------------|----------------|---------------------------|
| A | . - | ✓ | a |
| B | - ... | ✓ | |
| C | - . - . | ✓ | |
| D | - .. | ✓ | |
| E | . | ✓ | |
| F | .. - . | ✓ | |
| G | - - . | ✓ | |
| H | | ✓ | |
| I | .. | ✓ | |
| J | . - - - | ✓ | |
| K | - . - | ✓ | |
| L | . - .. | ✓ | |
| M | - - | ✓ | |
| N | - . | ✓ | |
| O | - - - | ✓ | |
| P | . - - . | ✓ | |
| Q | - - . - | ✓ | |
| R | . - . | ✓ | |
| S | ... | ✓ | |
| T | - | ✓ | |
| U | .. - | ✓ | |
| V | ... - | ✓ | |
| W | . - - | ✓ | |
| X | - .. - | ✓ | |
| Y | - . - - | ✓ | |
| Z | - - .. | ✓ | |

| Buchstabe | Morse Code | Wird dekodiert | Beim Trainer enthalten in |
|-----------|-------------|----------------|---------------------------|
| 1 | . - - - - | ✓ | 9 |
| 2 | .. - - - | ✓ | |
| 3 | ... - - | ✓ | |
| 4 | - | ✓ | |
| 5 | | ✓ | |
| 6 | - | ✓ | |
| 7 | - - | ✓ | |
| 8 | - - - .. | ✓ | |
| 9 | - - - - . | ✓ | |
| 0 | - - - - - | ✓ | |
| Ä | . - . - | ✓ | ä |
| Ö | - - - . | ✓ | |
| Ü | . - - - | ✓ | |
| CH | - - - - | ✓ | |
| ß | ... - - .. | | Nicht enthalten |
| – | .. - - . - | | |
| (| - . - - . | | |
|) | - . - - . - | | |
| " | . - . . - . | ✓ | |
| ; | - . - . . | ✓ | |
| ! | - . - . - - | ✓ | |
| ' | . - - - - . | ✓ | |
| . | . - . - . - | ✓ | ? |
| , | - - . . - - | ✓ | |
| : | - - - . . . | ✓ | |
| ? | .. - - . . | ✓ | |
| - | - - | ✓ | |
| = | - . . . - | ✓ | |

| Buchstabe | Morse Code | Wird dekodiert | Beim Trainer enthalten in |
|--|-------------------|----------------|---------------------------|
| / | - . . - . | ✓ | ? |
| @ | . - - . - . | ✓ | |
| + <AR> (Spruchende) | . - . - . | ✓ | ?<◇ |
| <KA> (Spruchanfang) | - . . - - | ✓ | ◇ |
| <KN> (Nur die gerufene Station möge antworten> | - . - - . | ✓ | |
| <AS> (Bitte warten) | . - . . . | ✓ | |
| <SK> (Verkehrsende) | . . . - . - | ✓ | |
| <VE> (Verstanden) | . . . - . | | <i>Nicht enthalten</i> |
| <SOS> (Notruf) | . . . - - - . . . | | |
| <HH> (Irrung) | | | |