PicATU500-Fernsteuerung
und
PicATU500v2 neue HW
SW-Version 2.xx

Andreas Lindenau DL4JAL

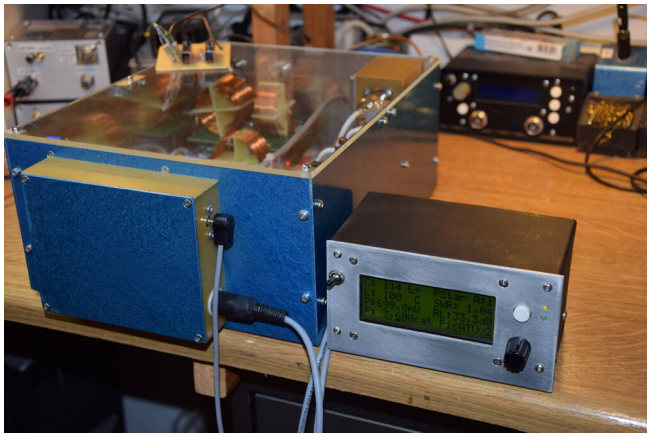
18. März 2024

Zusammenfassung

Der „PicATU500“ ist ein Bastelprojekt von mir. Der PicATU500 wurde mehrmals erfolgreich aufgebaut. In die neue Hardware „PicATU500v2“ sind alle neuen Erkenntnisse eingeflossen.

Die Bedienelemente am Tuner habe ich weg gelassen. Die Fernbedienung arbeitet so zuverlässig, das alle Funktionen über die Fernbedienung gemacht werden können.

Fast alle Leiterplatten wurden neu entworfen, mit dem Programm *Kicad 7.0*.



Die neue Hardware PicATU500v2 und die Fernbedienung.

Die Fernbedienung mit der Software Version 2.0. Unten rechts wurde die neue Tuner-HW erkannt. Das Display zeigt die Sendeleistung und die Anpassung an die Kunstantenne.

Inhaltsverzeichnis

Kapitel 1

Fernsteuer-Baugruppe

Die Fernsteuerbaugruppe ist in einem Gehäuse 15cm breit, 10cm tief und 7cm hoch eingebaut und passt mit diesen Abmaßen gut ins Shack.

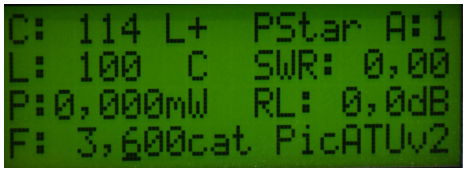
1.1 Software/Firmware

Grundsätzlich wird der Tuner über die Fernsteuerung betrieben!

Die Software wurde alles in Assembler geschrieben. In der Fernsteuer-BG habe ich einen PIC18F26K22 eingesetzt. Ab der SW Version 2.00 ist es nicht mehr möglich den PIC18F25K22 zu verwenden. Der Flashspeicher ist zu klein. Die Taktfrequenz beträgt 18,432MHz und wird mit einem Quarz erzeugt. Mit dem Quarz wird eine genauen Baudrate erzeugt. Der PIC18F26K22 hat zwei echte RS232 Schnittstellen. Die erste RS232 benutze ich für die Verbindung zum PicATU500v2. Mit der zweiten RS232 wird die CAT-Verbindung zum Transceiver hergestellt. Eine 3. RS232 habe ich per SW nachgebildet (nur TTL-Pegel). Damit wird mein Split-Filter (Frequenz-Info) gesteuert, so dass immer der richtige Tiefpass/Hochpass aktiv wird. Aber das ist nur für mich relevant.

1.1.1 Automatische HW-Erkennung des Tuner-Typ

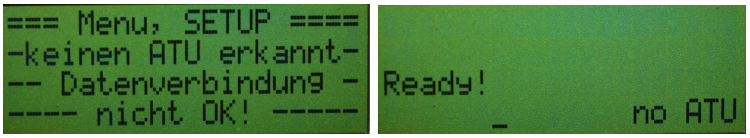
Ab der Version 1.06 der Firmware ist die HW-Erkennung des Tuners implementiert. Mit dieser Fernsteuerung ist es möglich auch meinen älteren sogenannten *SymTuner 750W mit Schrittmotor und RS232 Stromschleife* fernzusteuern. Auch die neu HW PicATU500v2 wird extra erkannt. Das SETUP wird entsprechend erweitert.



Nach Power-ON erfolgt eine Datensatz-Anforderung an den Tuner. Sendet der PicATU500 den Antwort-Datensatz steht in Zeile 4 rechts „PicATUv2“ für Pi-

cATU500 Hardware Version 2.0. Es ändern sich auch entsprechend die Struktur der Funktionen im Menü und im SETUP.

Keinen Tuner erkannt „no ATU“ Ist keine Datenverbindung angeschlossen oder funktioniert die Datenverbindung noch nicht, kann das *Menü* nicht aufgerufen werden. Ab FW 1.11 wird im Display eine entsprechende Info angezeigt.



Das linke Bild kommt wenn man mit der Einzeltaste versucht das *Menü* aufzurufen. Rechtes Bild ist das Display wenn noch keine Verbindung zum ATU aufgebaut ist. **(Anzeige ab FW 1.11)**

1.1.2 SETUP, “no ATU“

Ab FW 1.12 habe ich zusätzlich jetzt noch ein kleines SETUP programmiert. Somit kann die CAT-TRX-Schnittstelle getestet werden, auch ohne Datenverbindung zum PicATU500. Folgende Punkte sind im SETUP enthalten:

TRX Cat. select Auswahl des TRX. Siehe auch Kapitel 1.1.6.1 auf Seite 16.

Firmware Versionen? Aufruf der FW-Version.

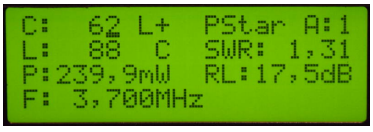
TRX-CAT Monitor, Hexadezimal Mit dem Monitor wird sichtbar wenn Infos über die CAT-Schnittstelle kommen. Alle Zeichen in HEX-Darstellung. Siehe auch Kapitel 1.1.6.7 auf Seite 21.

TRX-CAT Monitor, Ascii Mit dem Monitor wird sichtbar wenn Infos über die CAT-Schnittstelle kommen. Die Zeichen werden als String dargestellt. Siehe auch Kapitel 1.1.6.8 auf Seite 21.

Break!! Abbruch des SETUP.

1.1.3 Bedienelemente, LCD-Anzeige

Im Unterschied zur LCD-Anzeige im PicATU500-Tuner hat die Fernsteuer-Baugruppe ein LCD-Display mit 4x 20 Zeichen. Die Ausgabe aller Informationen wird dadurch sehr übersichtlich. An der Frontplatte ist rechts vom Display eine Einzeltaste und ein Drehgeber mit Taste für die Bedienung der Fernsteuerung.



Die Displayanzeige mit Sendesignal. Mit dem Display 4x20 Zeichen kann alles gut dargestellt werden. Ab FW 1.06 steht in Zeile 4 rechts der erkannte Tunertyp.

Normalbetrieb

Einzeltaste *kurzer Tastendruck* Mit einem kurzen Tastendruck der Einzeltaste kommen wir in das Menü.

Einzeltaste *langer Tastendruck* Der lange Tastendruck der Einzeltaste startet das **SETUP**.

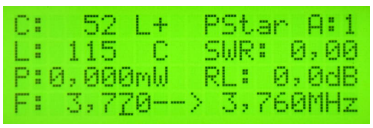
Taste im Drehgeber *kurzer Tastendruck* Der Kursor rückt um eine Stelle weiter. Je nachdem welcher Modus (Modus:Frequenzänderung oder Modus:LC-Änderung) aktiv ist.

Taste im Drehgeber *langer Tastendruck* Modus-Wechsel (Modus:Frequenzänderung oder Modus:LC-Änderung).

Drehgeber Je nach Modus und Kursorposition ändert sich die Frequenz oder die Werte im LC-Glied.

Drehgeber-Modus: Frequenzänderung

In Ausnahmefällen kann man die Frequenz im Display verändern. Je nachdem wo der Kursor steht wird auf- oder abwärts verstellt. Nach 2 Sekunden wird die neue Frequenz zum ATU gesendet. So ist es möglich einfach andere Frequenz im 10kHz-Raster auszuwählen. Erneute CAT-Befehle vom TRX überschreiben die Frequenz wieder.



Zeile4: Links steht die neue Frequenz der Fernbedienung, Rechts die Frequenz im ATU. Nach 2 Sekunden wird die neue Frequenz zum ATU gesendet. Der Tuner holt sich aus dem externen Eeprom eine neue gültige LC+LC-Variante aus dem Eeprom. Nach 5 Sekunden sehen wir an der Fernbedienung die neue Einstellung der Werte L,C und LC-Variante.

Drehgeber-Modus: LC-Änderung

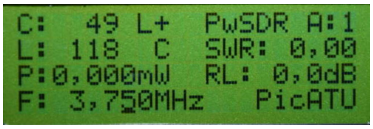
Nach einem *langen Drehgebertastendruck* wechselt der Kursor zum C-Wert/Einerstelle. Mit *kurzen Drehgebertastendruck* schaltet der Kursor weiter zur nächsten Stelle. Mit dem Drehgeber können wir die LC-Werte verstellen. Jede Änderung wird sofort zum ATU gesendet.

Diese Funktion wird kaum benötigt. Die automatische Abstimmung funktioniert so gut, dass eine manuelle LC-Änderung nicht notwendig ist.

1.1.4 Frequenzinformation und CAT-Anbindung des TRX

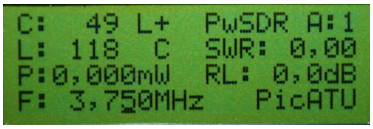
Soll die Tuner-Abstimmung aus dem Speicher des PicATU500 ausgelesen werden braucht der PicATU500 eine Frequenzinformation. Ich liste mal die verschiedenen Möglichkeiten der Frequenzinformation auf:

Frequenzmessung im Richtkoppler Die Frequenz wird im Richtkoppler des PicATU500 gemessen. Dazu ist ein Sendesignal notwendig. Der Pegel sollte so groß sein, dass ein sichere Frequenzmessung erfolgt. Einstellung im *SETUP „Set dBm Minimum“* (default = 22dBm, 158mW). Das funktioniert mit dieser geringen Leistung nur im unteren Frequenzbereich der Kurzwelle. Bei 28MHz sind etwa 12dB mehr Pegel nötig (etwa 34dBm, 2,5W). Die gemessene Frequenz ist sichtbar im Display der Fernsteuerung und des PicATU500. Je nach Sendefrequenz ändert sich die Frequenzanzeige.



Hinter Frequenz steht „MHz“.

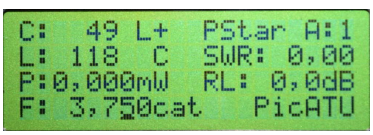
Frequenzmessung RK. oder Fernbedienung ohne TRX-Cat Die Frequenz kann auch an der Fernbedienung mit dem Drehgeber eingestellt werden. Siehe Kapitel 1.1.3. Die *Frequenzmessung im Richtkoppler* wird aber weiterhin benutzt für das Auslesen der Einstellungen aus dem Speicher. Sichtbar im Display Fernsteuerung und PicATU500. Je nach Sendefrequenz ändert sich die Frequenzanzeige. Je nach Frequenzeinstellung in der Fernsteuerung ändert sich die Frequenzanzeige.



Hinter Frequenz steht immer noch „MHz“.

Frequenz kommt vom TRX per CAT Das ist die dritte Möglichkeit der Frequenzinformation an die Fernsteuerung und den PicATU500. **Diese Variante ist die Beste**. Sobald einmal eine gültige Frequenz über die CAT-Schnittstelle vom TRX kommt, schaltet der PicATU500 um und es wird für das Auslesen der Einstellungen aus dem Speicher nur noch die CAT-Frequenz genommen. Die Anzeige im Display ändert sich im PicATU500 (von 3,75M in 3,75c) und der Fernbedienung (von 3,750MHz in 3,750cat). *Die Frequenzmessung im Richtkoppler wird nur noch am Beginn der Match-ReMatch-Befehle benutzt, für das Erkennen eines stabilen Sendesignales.*

Mit dem Drehgeber in der Fernbedienung kann die Frequenz noch geändert werden (funktioniert nicht bei jedem TRX) aber nicht mehr mit *der Frequenzmessung im Richtkoppler*. Die Frequenz im PicATU500 ändert sich parallel mit der Einstellung des VFO im TRX und damit auch die Match-Einstellungen im PicATU500.

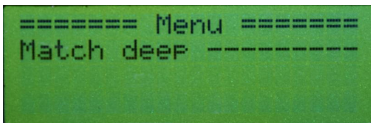


Die Anzeige der Frequenz ändert sich in der Fernbedienung „MHz“ wird durch „cat“ ersetzt.

1.1.5 Menübefehle, PicATU500 oder PicATU500v2

Als erstes beschreibe ich die Menübefehle, wenn als Tuner der *PicATU500v2* oder *PicATU500* erkannt wurde (*das ist der Normalfall*). Bei beiden Tunertypen ist die Struktur der Menübefehle gleich geblieben. Mit einem *kurzen Tastendruck, Einzeltaste* kommen wir in das *Menü* der verschiedenen Befehle. Diese Art der **Bedienung über die Fernsteuer-Baugruppe ist die Regel**. Mit dem Drehgeber wählen wir die einzelnen Menüpunkte aus und starten die Menüpunkte mit der Einzeltaste. Neu ab FW 1.14, die letzte Funktion merkt sich die FW und die Menüpunktnummer wird im Eeprom gespeichert, so dass auch nach *PowerON* der Menüpunkt als erstes vorgelegt wird. **Vorrang hat nur die SETUP-Einstellung im Kapitel 1.1.6.6 auf Seite 21, wenn sie aktiviert wurde.**

1.1.5.1 Match, Match deep



In der Beschreibung habe ich beide Funktionen *(Match, Match deep)* zusammen gefasst. Wie ich schon erwähnt habe, wurde die „Match-Funktion“ von mir noch einmal grundlegend überarbeitet. Ich habe auf eine intelligente Flächensuche umgestellt. Die Fläche ergibt sich aus den Seitenlängen X und Y. In unserem Fall ist X=C-Werte 0 bis 2047 und Y=L-Werte 0 bis 255. Das ist eine Fläche von 2047 x 255 Feldern, das ergibt multipliziert 512985 Felder. Das ist eine enorme Menge. Deshalb bin ich auf die Idee gekommen *die erste Flächensuche mit exponentieller Schrittweite* zu beginnen.

Nachdem das Sendesignal erkannt wurde beginnt die erste Such-Funktion *Grundmatch*. Das ist die Wichtigste. Die Schrittweite steigt quadratisch und es wird immer nur ein Relais im Glied der binären L- oder C Kette eingeschaltet. Displaybilder der Match-Funktion sind im Kapitel 1.1.5.2 auf Seite 10 zu sehen.

Grundmatch Wie schon erwähnt ist das die erste Suche des SWR-Minimums. Die Suche beginnt mit C Wert= 0 und L Wert=0. Es folgt der nächste Schritt mit C-Wert=8 und L-Wert=1. Bei jedem weiteren Schritt verdoppelt sich der Wert. Es ergeben sich 9x9 also 81 L/C Kombinationen. Die L- und C-Werte kann man als eine quadratische Fläche betrachten mit 81 Feldern.

L-Werte der Y-Achse 0, 1, 2, 4, 8, 16, 32, 64, 128

C-Werte der X-Achse 0, 8, 16, 32, 64, 128, 256, 512, 1024

	C	0,0pF	12,5pF	25pF	50pF	100pF	200pF	400pF	800pF	1600pF
	C-Glied	0	8	16	32	64	128	256	512	1024
L	L-Glied									
0,00uH	0									
0,125uH	1									
0,25uH	2									
0,5uH	4									
1uH	8									
2uH	16									
4uH	32									
8uH	64									
16uH	128									

Die Wertigkeit von L geht von 0 (0,00uH) bis 128 (16uH), beim PicATU500 geht die Wertigkeit von C von 0 (0,0pF) bis 1024 (1600pF) und beim PicATU500v2 geht die Wertigkeit von C von 0 (0,0pF) bis 1024 (800pF). Die Suche auf der Fläche (gelb) beinhaltet 81 Felder. Bei jeder Kombination von L und C wird das Return-Loss gemessen. Das Feld mit dem höchsten Return-Loss merkt sich die Funktion. Nachdem *Grundmatch* beendet wurde, wird eine neue Suchfläche (hier grün) festgelegt. Es beginnt die Suchfunktion *Submatch*. Die Schrittweite in der grünen Fläche wird neu gewählt, so das wieder etwa 64 bis 81 Suchfelder entstehen.

Submatch *Submatch* setzt die Suche fort mit passender Schrittweite in der quadratischen Fläche von L und C. Die Suche geht in unserem Beispiel C-Werte von 64 bis 256 und L-Werte von 8 bis 32 (oben grünes Feld). Die Schrittweite wird passend eingestellt. In unserem Beispiel C-Step=32 und L-Step=4. Mit der neue Schrittweite wird noch einmal die Suchfläche aktualisiert.

C-Wert	64	96	128	160	192	224	256
L-Wert							
8							
12							
16							
20							
24							
28							
32							

Jetzt ist die Schrittweite konstant linear. C Step 32 und L Step 4. Im Beispiel wurde neues RL Maximum ist gefunden (Rot, L=16, C=96).

Wiederholung der Funktion *Submatch* mit reduzierter LC-Fläche und reduzierten Step. Vor jedem erneuten Submatch wird die Schrittweite halbiert.

C-Wert	64	80	96	112	128	144	160
L-Wert							
8							
10							
12							
14							
16							
18							
20							
22							
24							

Jetzt ist die Schrittweite halbiert.C Step 16 und L Step 2. Ein neues RL Maximum ist gefunden (Rot, L=16, C=112).

Wiederholung der Funktion *Submatch* mit reduzierter LC-Fläche und reduzierten Step.

C-Wert	88	96	104	112	120	128	136
L-Wert							
12							
13							
14							
15							
16							
17							
18							
19							
20							

Die Schrittweite wird wieder halbiert. Bei C Step 8 und bei L Step 1. Ein neues RL Maximum ist gefunden (Rot, L=15, C=104).

Die Schrittweite wird immer wieder halbiert und mit der Funktion *Submatch* neu gesucht. **Die Funktion *Submatch* wird beendet, wenn:**

- Beide Schrittweiten für L und C sind 1. Die beste L/C Kombination wird gemerkt und verwendet. Ist dabei das SWR kleiner 1,5 wird diese Einstellung im 10kHz-Raster gespeichert.
- Das SWR ist kleiner 1,2. Der Match-Vorgang war erfolgreich. Die L/C Kombination wird gemerkt und verwendet.

So finde ich ganz schnell eine Impedanzanpassung. Es dauert im Idealfall etwa 10 bis 15 Sekunden.

Besonderheit bei „Match“: Ist bei der ersten Matrix-Suche *Grundmatch* in der L/C Variante das SWR $< 2{,}5$, wird in dieser L/C Variante das SWR Minimum gesucht. Ansonsten wird sofort die nächste L/C Variante gewechselt, beginnend mit *Grundmatch* usw...

Besonderheit bei „Match deep“: Die Schwelle SWR < 2,5 ist außer Kraft. Es wird immer in jeder L/C-Variante *in die Tiefe* gesucht.

Bei beiden Match-Varianten wird zuerst angenommen, dass die Antennenimpedanz > 50 Ohm ist, wie das meistens der Fall ist. **Deshalb habe ich folgende Suchreihenfolge programmiert:**

- **Variante1:** L/C Glied Impedanz größer 50Ohm, Grundmatch, Submatch....
- **Variante3:** C/L Glied Impedanz größer 50Ohm, Grundmatch, Submatch....
- **Variante2:** L/C Glied Impedanz kleiner 50Ohm, Grundmatch, Submatch....
- **Variante4:** C/L Glied Impedanz kleiner 50Ohm, Grundmatch, Submatch....
- Ist bei einer Varianten das beste SWR > 2,5, wird sofort in der nächsten Variante gesucht. Bei *Match deep* wird nicht bei SWR > 2,5 abgebrochen, sondern bis zum Ende weiter gesucht. Ende bei Schrittweite L=1 und C=1 oder das SWR ist kleiner 1,2.
- Wird mit *Match* keine Anpassung gefunden kann man mit *Match deep* die Suche der Anpassung wiederholen.
- Ein „Match deep“ bricht nicht mehr bei „SWR > 2,5“ ab, sondern sucht immer bis zum Ende (bis beide Schrittweiten 1 sind). So ist die Suche nach dem besten SWR intensiver. Es wird fast immer ein vertretbares SWR gefunden.
- Ist das SWR < 1,5, werden die Werte von L, C und die L/C Variante im Speicher des „10kHz-Segmentes“ abgelegt. Zusätzlich kann man noch die gefunden Einstellung mit dem Befehl *(B)and* für das ganze KW-Band abspeichern (in alle 10kHz-Segmente des Bandes).

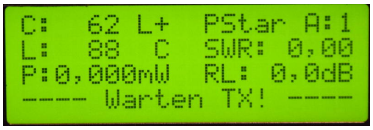
Wie schon oben kurz erwähnt suchen die Funktionen *Match* und *Match deep* bis das SWR den Wert 1,2 unterschreitet. Dann bricht die Funktion ab mit „Abstimmung OK“. Mit *ReMatch* kann man anschließend die Suche fortsetzen und ein noch besseres SWR finden.

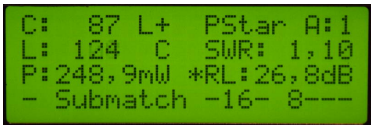
1.1.5.2 ReMatch

Im Unterschied zur Funktion *Match, Match deep* bricht die Funktion *ReMatch* **nicht** bei SWR kleiner 1,2 ab, sondern sucht so lange bis alle Felder durchsucht wurden. Ist die kleinste Schrittweite L=1 und C=1 erreicht wird *ReMatch* beendet.

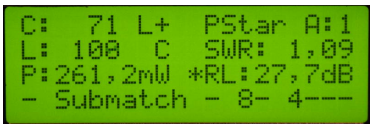
Beispielbilder von „ReMatch“ Hier noch ein Beispiel einer Nachabstimmung. Mit allen Zwischen-Display-Bildern.

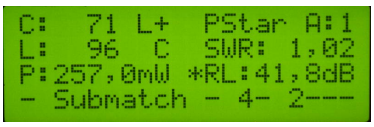
Zuerst die Kontrolle wie das SWR vor *ReMatch* war. Im Menu den Befehl *ReMatch* starten.



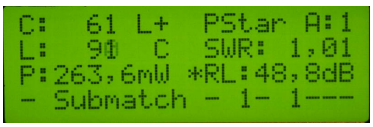


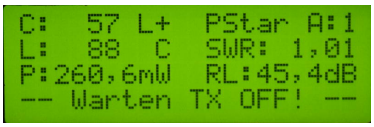
Der PicATU500 wartet auf das Sendesignal. Die Suche beginnt in einem größeren LC-Feld. Das erste Suchergebnis *SubMatch mit Schrittweite L und C* wird im Display angezeigt.



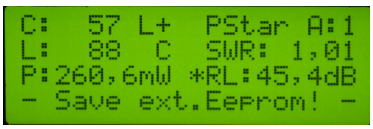


Nach jeden *SubMatch* wird ein Zwischen-Datensatz übermittelt. Das Display wird aktualisiert. Das SWR wird immer besser und die Schrittweiten kleiner.



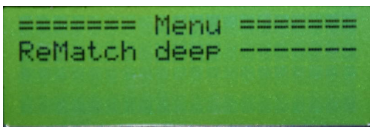


Ein gutes SWR. Ein super Ergebnis. Die Schrittweite ist beim kleinsten Wert angekommen. Anschließend wird mit großer Messzwischenzeit noch einmal im MatrixLC 4x4 nachgestimmt. Return Loss ändert sich noch etwas. Am „Match ENDE“ wartet der Tuner bis der Sender AUS ist.



Das Match-Ergebnis wird im Eeprom abgelegt. Die Messergebnisse werden anschließend noch 15 Sekunden angezeigt.

1.1.5.3 ReMatch deep

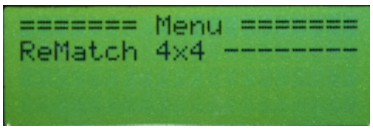


Die Funktion *ReMatch deep* sucht eine Anpassung in der eingestellten LC-Variante von Anfang an mit Grundmatch, Submatch usw., wie in der Funktion *Match* beschrieben. Diese Funktion habe ich zusätzlich programmiert für die ganz „hartnäckigen Fälle“. *ReMatch deep* ist die *Match deep* Funktion ohne Wechsel der LC-Varianten. Auch hier endet die Suche erst wenn beide Schrittweiten 1 sind.

Neu ab der FW 1.13 im Tuner: Wird bei „ReMatch“ oder „ReMatch Deep“, Schrittweite größer 1/1 schon ein gutes SWR $< 1{,}1$ gefunden, wird sofort mit Schrittweite 1/1 in einer Fläche L/C 4x4 die Suche fortgesetzt und Abgeschlossen.

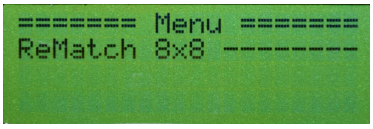
Der Abstimmvorgang geht so schneller, teilweise erheblich schneller.

1.1.5.4 ReMatch 4x4



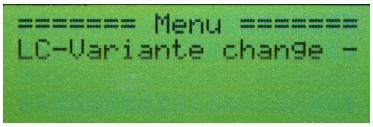
Von den momentanen L- und C-Werten aus gehend werden +/− 2 Werte auf der X-Achse angenommen und +/− 2 Werte auf der Y-Achse. Es ergeben sich (4x4) 16 Suchfelder die der Reihe nach eingestellt werden und das *Return Loss* verglichen wird. Das Suchfeld mit dem höchsten *Return Loss* wird anschließend eingestellt. Die Zwischenzeit zwischen den *Return Loss* Messungen wird auf 100 mSek. erhöht, für genauere Messungen. Die Abschlussmessung mit den Ergebnissen (bleiben 15 Sekunden im Display stehen).

1.1.5.5 ReMatch 8x8



Aus gehend den momentanen L- und C-Werten werden +/− 4 Werte genommen. Es ergeben sich (8x8) 64 Suchfelder, in denen Reihe nach das größte *Return-Loss* gesucht wird. Das Suchfeld mit dem höchsten *Return-Loss* wird anschließend eingestellt. Die Zwischenzeit zwischen den SWR Messungen wird etwas erhöht (40mSek.), für genauere Messungen. Die Abschlussmessung mit den Ergebnissen (bleiben 15 Sekunden im Display stehen).

1.1.5.6 LC-Variante



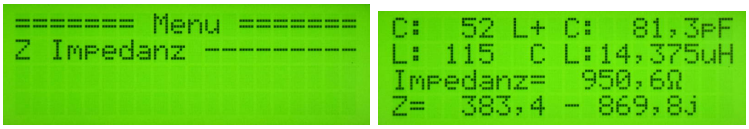
Diese Funktion kann die L/C Variante gezielt ändern. Im LCD-Display werden die LC-Varianten angezeigt. Die entsprechende LC-Variante wählen und mit der Einzeltaste bestätigen. Die Funktion *LC-Variante* braucht man, wenn in der Variante *nur-L* und *nur-C* nach einer Impedanzanpassung gesucht werden soll. Anschließend könnte die Funktion *ReMatch* oder *ReMatch deep* gestartet werden.

Hier im Beispiel habe ich an einer hochohmigen Antennen die LC-Variante gewechselt und anschließend mit *ReMatch* das SWR optimiert. Oft verbessert sich das SWR.

1.1.5.7 Z Impedanz

Es wird die Impedanz der Antenne angezeigt, zuerst der Betrag und dann noch Komplex. Wurde ein gutes SWR gefunden ist es ja kein Problem per Komplexer Berechnung rückwärts die komplexe Impedanz der Antenne für diese Frequenz zu berechnen. Allerdings gehen bei höheren Frequenzen die komplexen Werte des Balun's mit ein. Da können die Berechnungen sehr abweichen von der wirklichen Impedanz. Ab der Firmware 1.06 findet die Berechnung der Impedanz in der Fernsteuer-Baugruppe statt und nicht mehr im Tuner. Für den Anwender spielt das aber keine Rolle. Ich konnte dadurch im Tuner Flash-Speicher einsparen, so das auch ein PIC18F4520 oder PIC18F45K22 mit kleinerem Flash-Speicher einsetzbar ist. Das gilt aber nur für den Tuner PicATU500. Im PicATU500v2 Hardware V2.0 wurde von mir der PIC18F25K22 eingesetzt.

Es soll ja nur eine zusätzliche Information sein.

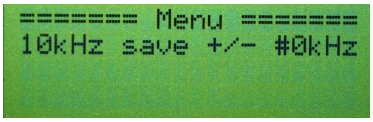


Zusätzlich zur Impedanz wird auch die Größe des LC-Gliedes (in uH und pF) mit angegeben und welche LC-Variante geschaltet ist.

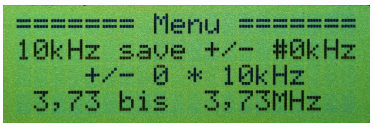
1.1.5.8 Band save

Die gefundene Impedanzanpassung wird in allen 10kHz Segmenten des AFU-Frequenz-Bandes abgespeichert und noch etwas darüber hinaus. Das erleichtert das Nachstimmen der Impedanzanpassung erheblich, da immer ein Ausgangswert für *ReMatch* im Speicher vorhanden ist.

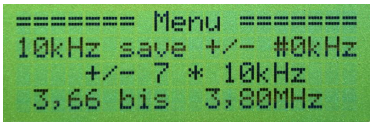
1.1.5.9 10kHz save +/− #0kHz



Die gefunden Relaiseinstellung wird im 10kHz Segment der aktuellen VFO-Frequenz abgespeichert. Je nach Einstellung mit dem Drehgeber ist der Speicherbereich von +/− 0kHz bis +/− 90kHz möglich. Der Frequenzbereich steht in Zeile 4.



Ein Frequenz-Beispiel, Auswahl #0: die Frequenz beträgt 3,730 000 MHz: die Relaiseinstellung wird im 10kHz Segment 373 abgespeichert.

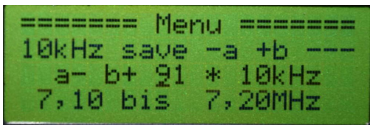


Noch ein Beispiel, Auswahl #7: die Frequenz beträgt 3,730 000 MHz: die Relaiseinstellung wird in $+/-$ 7 $*$ 10kHz Segment abgespeichert. So wie das im Display Zeile 4 steht.

1.1.5.10 10kHz save $-a+b$

Diese Funktion ist ähnlich der Funktion im vorherigen Kapitel. Es wird die Relaiseinstellung der aktuellen Frequenz im Segment 10kHz gespeichert. Zusätzlich kann in den Nachbarsegmenten (bis zu 9 Segmente) die Einstellung mit gespeichert werden. In dieser Funktion kann man eine getrennte Einstellung für die Nachbarsegmente nach unten und die Nachbarsegmente nach oben wählen. Die Anzahl der 10kHz-Segmente kann unterschiedlich sein. Mit der Taste im Drehgeber wird der Kursor umgeschaltet und die Einstellung gewechselt.

Im Display sehen wird die Frequenz der 10kHz-Segmente *von/bis*.



Die Sendefrequenz beträgt 7,190MHz. Von dieser Frequenz aus wird die Relaiseinstellung in 9 10kHz-Segmenten nach unten und 1 10kHz-Segment nach oben mit gespeichert, von 7,100 MHz bis 7,200MHz.

1.1.5.11 Band select

Diese Funktion hat nur einen Sinn, wenn keine CAT-Steuerung erfolgt und möglichst schnell auf eine andere Bandfrequenz geschaltet werden soll. Ich habe diese Funktion noch nicht benötigt.

1.1.5.12 LCD L+C Zahl/Werte

Diese Funktion ist neu ab FW 2.03. Bisher wurde im LCD-Display immer nur der Wert von L und C als eine Zahl dargestellt. Wenn man aber die binäre Zahl als Kapazität in pF oder Induktivität in uH sehen wollte, ging das über die Funktion „Z Impedanz“. Jetzt kann der Wert von C und L auch im normalen Betrieb dauerhaft auf dem Display dargestellt werden. Die Streukapazität (Angabe im SETUP) wird mit eingerechnet.

Links der Menüpunkt zum Umschalten, rechts Display **L als uH** und **C als pF** und nicht mehr die *Zahl 0..255 bei L* oder *Zahl 0..2047 bei C*.

1.1.5.13 Break!!, CLR Messwerte

Break!! ist nicht nur Abbruch sondern löscht auch sofort die angezeigten Messwerte im Display.

(ab FW 1.13) Wenn sich das Frequenzsegment (10kHz-Segment) ändert, entweder durch CAT-TRX oder Drehgeber, werden die Messwerte im Display gelöscht.

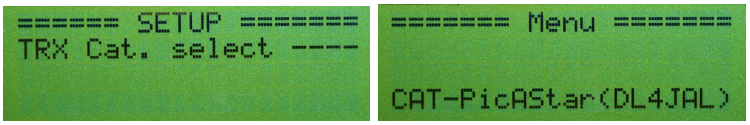
1.1.6 SETUP, PicATU500 und PicATU500v2

Als erstes beschreibe ich die SETUP-Befehle, wenn als Tuner der *PicATU500* oder *PicATU500v2* erkannt wurde. Das ist der eigentliche Normalfall.

Der *lange Tastendruck* der Einzeltaste führt in das SETUP. Im SETUP befinden sich Funktionen die nicht so oft gebraucht werden. Mit dem Drehgeber wählen wir die entsprechende Funktion aus. Die Einzeltaste startet die Funktion.

1.1.6.1 TRX Cat. select

Diese Funktion wählt den Typ des TRX aus.



Auf der Platine mit PIC18F26K22 ist noch eine 2. RS232-Schnittstelle für die TRX-CAT-Verbindung vorhanden. Auf der Platine sind 4 gleichberechtigte 5-polige Steckerbuchsen für verschieden HW-Varianten. Die Hardware der RS232-Schnittstellen müssen ja zum HW-Protokoll des TRX passen.

Folgende HW-Schnittstellen habe ich vorgesehen:

5V RX-TTL Diese Schnittstelle hat in beiden Richtungen TTL-Pegel 5V entspricht dem HIGH-Pegel und 0V entsprechen dem LOW-Pegel. Mein TRX PicAStar und vom Icom die CI-V Schnittstelle arbeitet ebenfalls mit TTL-Pegel.

RS232 Eine kleinen Platine mit einem RS232-IC (z.B.:MAX232) bildet die echte RS232-Schnittstelle mit einem SUB-D9 Stecker oder Buchse. Je nach dem wie es gewünscht wird. Ich habe einen weiblichen SUB-D9. Bei dieser HW ist etwa -8V der HIGH-Pegel und +8V der LOW-Pegel.

USB extern Die USB-HW ist ebenfalls auf einer kleinen Platine mit dem IC FT232RNL.

Die 4 Stecker J5 bis J8 arbeiten parallel. Es darf jeweils nur eine Möglichkeit benutzt werden, sonst gibt es „Datensalat“. Alle Steckplätze sind gleichberechtigt Die MC-Platine wurde auch mit Kicad 7.0 neu entworfen. **Hier noch mal die zwei verschieden Module:**

echte RS232 mit DB9 Buchse Das ist eine echte RS232-Schnittstelle, die einige TRX benötigen. Mit einem 9-poligen Stecker und einem IC *MAX 232 CWE* im SMD SOL16 Gehäuse. Also etwas breiter als normal.

echte USB mit FT232RNL Der FT232RNL ist eine virtuelle USB-RS232 in Richtung PC. Der IC FT232RL befindet sich jetzt nicht mehr auf der Hauptplatine sondern wurde in dieses USB-Modul ausgelagert.

In der Software habe ich folgende TRX berücksichtigt:

PicAStar(DL4JAL), TTL mit 3,5mm Stereo Klinkenbuchse Der CAT-Anschluss ist ein Ausgang mit 5V TTL-Pegel. Bei jedem VFO-Frequenzwechsel wird sofort die neue Frequenz ausgegeben.

Yaesu FT847, RS232 mit SUBD9 Buchse 9600-8N-2 Anschluss für den FT847

Icom CI-V Remote, TTL mit 3,5mm Stereo Klinkenbuchse für den Icom CI-V Anschluss

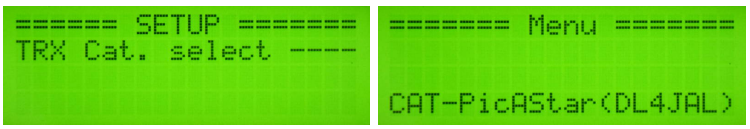
K2/K3 Elecraft, RS232 mit SUBD9 Buchse Funktioniert erst ab FW 1.13 richtig. Baudrate ist fest eingestellt 4800 Baud.

PC PowerSDR, USB-Buchse oder RS232 Für Windows-Software „PowerSDR“.

Yaesu FTDX101, RS232 mit SUBD9 Buchse Anschluss für FTDX101. Einstellung 9600Baud.

Yaesu FT1000MP Field, RS232 mit SUBD9 Buchse Anschluss FT1000MP-Field, 4800Baud.

Yaesu FT1000MP, RS232 mit SUBD9 Buchse Anschluss FT1000MP, 4800Baud.



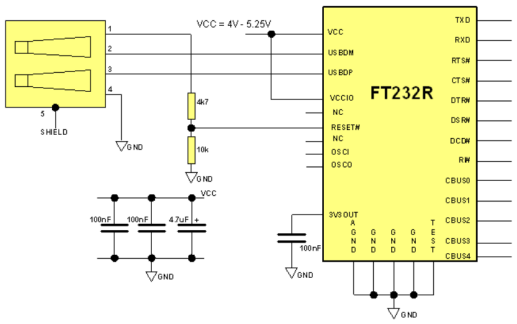
Als erstes kommt die aktive Einstellung. Bei mir ist das mein SDR-Homemade TRX PicAStar.

Die Auswahl mit dem Drehgeber. Mit der Einzeltaste bestätigt. Ein *langer Tastendruck* führt zum Abbruch der Funktion.

Beschreibung der „PowerSDR“ Schnittstelleneinrichtung (PC-Software)
Zuerst stecken wir die Verbindung mit USB-Kabel „Remote-BG“ und „PC“. Im PC wird eine neue serielle Schnittstelle sichtbar. Jetzt machen wir im „PowerSDR“ folgende Einstellungen. Im „SETUP, CAT Control“ wählen wir die neue Schnittstelle aus (9600,none,8,1). Es folgt noch **ID as** „PowerSDR“ und bei „Allow Kenwood AI Command“ setzen wir den Haken. Jetzt noch „Enable CAT“ Haken setzen und mit „OK“ bestätigen. PowerSDR sendet jetzt bei jeder Frequenzänderung ein Kommando zu unserer Remote-BG.

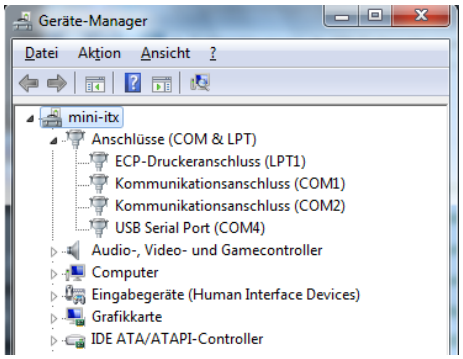
Die Inbetriebnahme der USB-Schnittstelle gestaltet sich manchmal etwas schwierig. Hier einige Tips dazu. Ich habe eine zusätzliche Beschaltung des IC FT232RL vorgenommen.

6.2 Self Powered Configuration

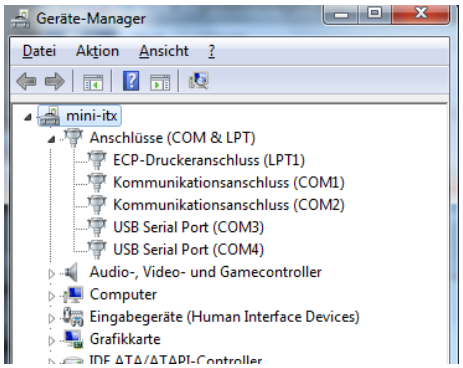


Die Beschaltung des FT232RL, wenn die Versorgungsspannung Baugruppe und 5 Volt USB getrennt sind. Der FT232RL wird beim anstecken des USB-Kabels neu gestartet (Reset).

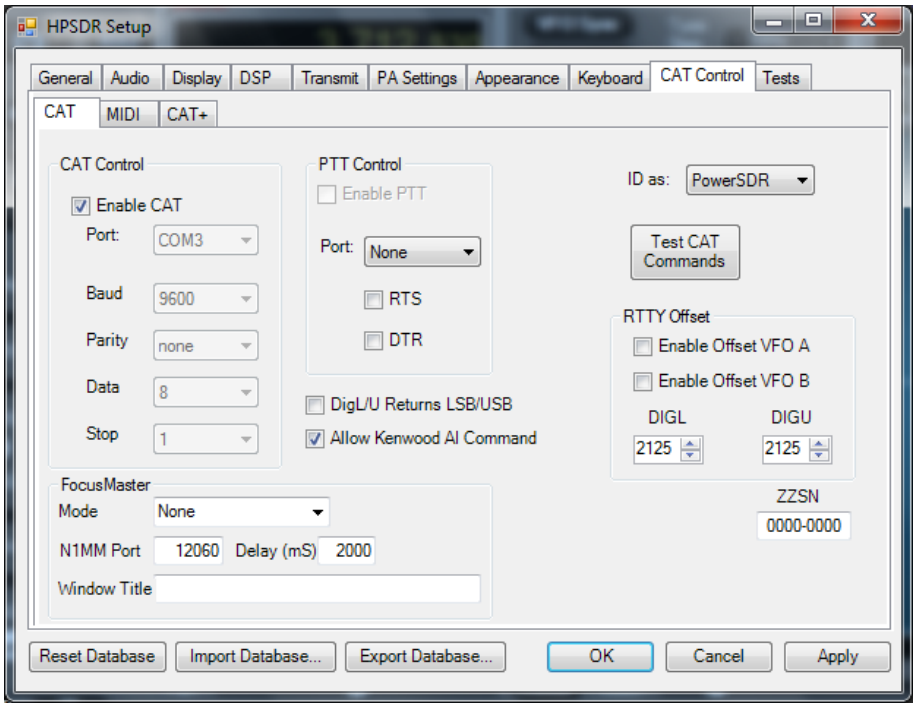
Als erstes öffnen wir am PC die „Systemsteuerung Gerätemanager“. Wird das USB-Kabel am PC angesteckt entsteht eine neue virtuelle RS232. Im „Gerätemanager“ ist das Online sichtbar. Als Test können wir das USB-Kabel mehrmals anstecken und wieder abziehen.



Das USB-Kabel ist noch nicht angesteckt. COM1, COM2 und COM4 sind zu sehen.



Das USB-Kabel wurde angesteckt. Wir sehen die neu entstandene COM. In diesem Fall die COM3.



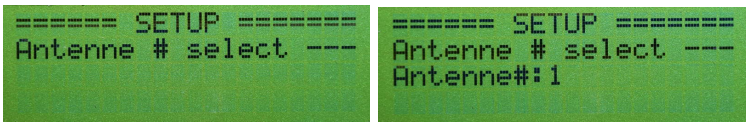
Jetzt können wir im „PowerSDR“ die neu entstandene Schnittstelle im Dialogfenster einstellen und benutzen. Wir sehen alle „CAT Einstellung“ im „PowerSDR“.

Auf der MC-Platine wird mit den Dioden D3, D4 und D8 alle ankommenden TX-Daten entkoppelt und können ohne Umschaltung parallel genutzt werden. Es darf aber nur immer eine Schnittstelle aktiv sein sonst gibt es „Datensalat“.

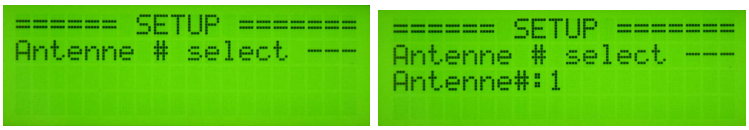
Mit dem „TRX-CAT Monitor“ im SETUP kann man überprüfen ob Daten vom TRX kommen.

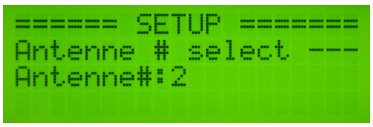
Wichtig ist, es darf nur eine TRX-Schnittstelle aktiv sein.

1.1.6.2 Antenne # select



Im PicATU500, Mikrocontroller-Platine befindet sich der Speicher-IC für alle gefundenen Match-Einstellungen von L,C und LC-Varianten. Verwendet habe ich einen seriellen Eeprom vom Typ 24LC512. Dieser Eeprom hat eine I^2C Schnittstelle zum Schreiben und Lesen der Daten. Das Speichervolumen beträgt 512 kBit. Das entsprechend 64 kByte Speicherplatz. Das ist so viel, dass es möglich ist für den Frequenzbereich 1,5 MHz bis 30 MHz aller 10kHz einen Speicher für L,C und LC-Variante vorzusehen.





Die Auswahl einer neuen Antennen-Nummer schaltet auf einen anderen Speicher-Adressbereich im externen Eeprom um.

Für einen Bereich von 1,5 MHz bis 30 MHz benötige ich 11400 Byte-Speicherplatz, pro Match-Einstellung sind 4 Byte nötig. Ich habe aber im Eeprom 65535 Byte Speicherplatz. Da ist immer noch viel übrig. Ich habe deshalb 5x den Speicherplatz von 1,5 MHz bis 30 MHz reserviert. 5x 11400 Byte sind 57000 Byte. Ich wusste nicht wie ich die einzelnen Speicherbereiche nennen soll. Da ist mir der Begriff „Antenne 1 bis 5“ eingefallen. Jede Antennennummer ist ein Speicherbereich 1,5 MHz bis 30 MHz (11400 Byte).

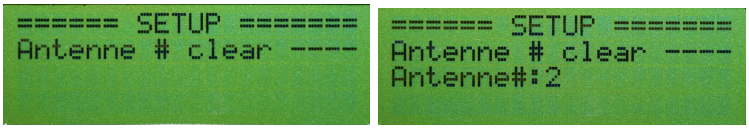
Antenne#	Adressbereich	HEX
1	0 bis 11400	0 – 0x2C88
2	11401 bis 22800	0x2C89 – 0x5910
3	22801 bis 34200	0x5911 – 0x8598
4	34201 bis 45600	0x8599 – 0xB220
5	45601 bis 57000	0xB221 – 0xDEA8

Welche Adressbereiche im Eeprom festgelegt sind ist eigentlich uninteressant. Die Tabelle dient nur zur Information, wie alles zusammen hängt.

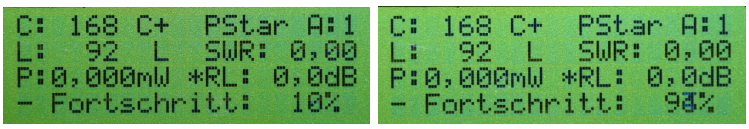
Welche Antennen-Nummer (Speicherbereich) aktiv werden soll wird in dieser Funktion festgelegt. Verschiedene Antennen-Nummern sind vorteilhaft beim Testen anderer Antennen. So kann man die einmal gefundenen Einstellungen und Werte erhalten. Möchte man eine neue Antenne testen, braucht man nur auf eine andere Antennen-Nummer schalten. Oder eine andere Idee ist eine Antennen-Nummer für trockenes Wetter und eine neue Antennen-Nummer für nasses Wetter.

Ein *langer Tastendruck* führt zum Abbruch der Funktion.

1.1.6.3 Antenne # clear



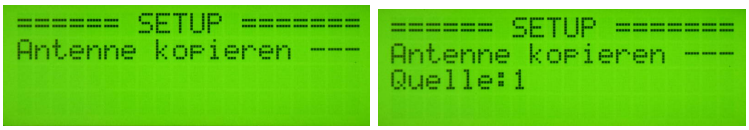
Die Funktion *Antenne # clear* löscht alle Speicherzellen für den Adressbereich der entsprechenden Antennen-Nummer. Ein Rückmeldung in Prozent erfolgt über die Fernsteuerung während des Löschvorgangs. Das Löschen dauert etwas länger. Ein *langer Tastendruck* führt zum Abbruch der Funktion.



Die Funktion dauert mehrere Minuten. Deshalb die Prozent-Information. Die Anzeige des *Fortschritts* ist in der neuen FW etwas anders, nur in Zeile 4.

1.1.6.4 Antenne kopieren

Mit *Antenne kopieren* wird der Speicherinhalt der Antenne „Quelle“ in den Speicherbereich der Antenne „Ziel“ kopiert. Auch hier wird der Fortschritt in Prozent angezeigt.



Anschließend wird noch die Zielnummer angegeben. Das kopieren beginnt. Auch wieder die Fortschrittsanzeige mit Prozent schließt sich an.

1.1.6.5 Firmware Versionen?

In dieser Funktion wird an der Fernbedienung die Version der Firmware abgefragt. Die Version im PicATU500 und die Version in der Remote-BG-Fernbedienung werden angezeigt.

1.1.6.6 SET Menu# PowerON –

Diese Funktion ist neu ab FW Version 1.14. Hier kann man den Menüpunkt festlegen, der als erstes nach PowerON auf dem Display angeboten wird. Im zweiten Schritt kann diese Funktion *aktiviert* oder *deaktiviert* werden. Ist die Funktion *deaktiviert* merkt sich die FW die zuletzt benutzte Funktion und legt diese vor nach kurzem Tastendruck der Taste 1. Ist die Funktion *aktiviert* wird nach *PowerON* und kurzem Tastendruck der Taste 1 die ausgewählte Menü-Funktion vorgelegt.

1.1.6.7 TRX-CAT Monitor, Hexadezimal

Dieser Monitor überwacht die RS232 zum TRX und schreibt jedes empfangene Byte als HEX-Zahl in die LCD-Anzeige.

Bei manchen TRX sind die Datensätze ein String: z.B.:

„FA00014320150;“

Der Datensatz besteht nur aus darstellbaren Zeichen. In Hexadezimal würde der gleiche String so aussehen:

„46 41 30 30 30 31 34 33 32 30 31 35 30 3B“

Jeder Buchstabe/Zeichen ist eine HEX-Zahl.

1.1.6.8 TRX-CAT Monitor, ASCII

Dieser Monitor überwacht die RS232 zum TRX und schreibt jedes empfangene Byte als ASCII-Zeichen in die LCD-Anzeige. Bei manchen TRX sind die Datensätze ein String: z.B.:

„FA00014320150;“

Der Datensatz besteht nur aus darstellbaren Zeichen. In Hexdezimal würde der gleiche String so aussehen:

„46 41 30 30 30 31 34 33 32 30 31 35 30 3B“

Jeder Buchstabe/Zeichen ist eine HEX-Zahl.

1.1.6.9 TRX-CAT Monitor, Funktion der Tasten

Taste 1, Einzeltaste Monitor beenden. Die FW startet neu.

Taste 2, Taste im Drehgeber LCD löschen. Zusätzlich bei TRX:

FT847 Daten senden „Open RS232“ und anschließend „Get Frequenz“.

Elecraft K2/K3 Befehl senden „AI2“. Daten kommen dann automatisch ohne Anforderung.

FTDX101 Befehl senden „FA;“. Daten kommen zurück.

FT1000MP Field Befehl senden „Get CAT Status“. 5 Byte kommen zurück.

FT1000MP Befehl senden „Get CAT Status“. 5 Byte kommen zurück.

1.1.6.10 SET C-Platine, Wert KC11, Streukapazität

Die alte Platine für das C-Glied hat die größte Kapazität KC11 von 1600pF und die kleinste Abstufung beträgt 1,6pF. Die neue Platine für das C-Glied hat als größte Kapazität KC11 von 800pF. Dadurch ergibt sich eine Schrittweite von 0,8pF. Den Wert für KC1 müssen wir hier einstellen, damit die Impedanz der Antenne richtig berechnet werden kann. Die zweite Einstellung ist die Restkapazität des C-Gliedes wenn alle Relais abgefallen sind. Sie beträgt etwa 27pF. Auch dieser Wert geht mit in die Berechnungen ein. **Beide Einstellungen haben aber keine Auswirkungen beim Tunen.**

1.1.6.11 Frequenz manuell, CAT aktivieren

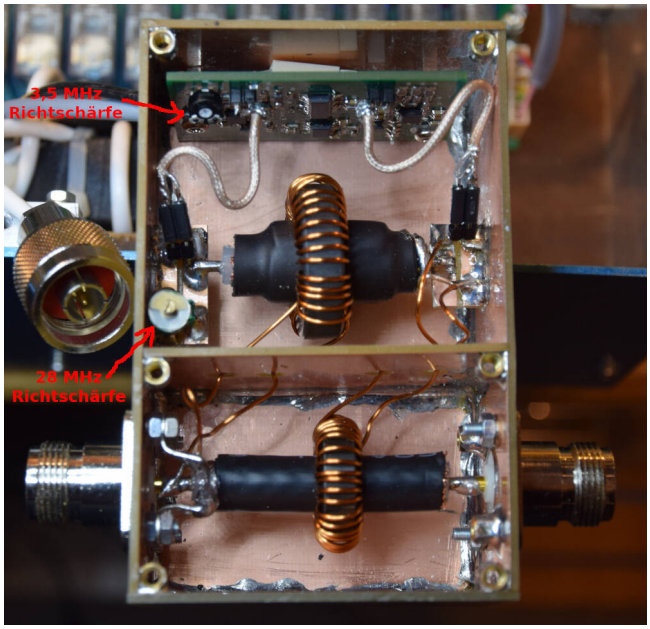
Diese Funktion ist neu ab Version 2.0 für beide Tunertypen. Wenn der Tuner mit einen Transceiver betrieben wird der keine CAT-Anbindung an die Fernbedienung hat, zum Beispiel ein QRP-Transceiver Eigenbau, kann man hier einstellen das die Frequenzmessung im Tuner deaktiviert wird und nur die Frequenzeinstellung von der Fernbedienung gültig ist. **Im Default ist dieser Punkt aktiviert**.

1.1.7 SETUP, PicATU500v2 neu hinzu gekommen

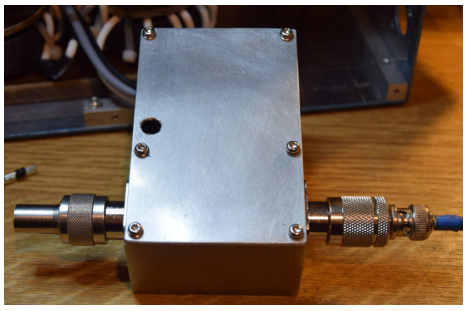
Für die neue Hardware Version 2.0 PicATU500v2 sind noch einige SETUP-Funktionen hinzu gekommen. Alle SETUP-Funktionen, die ursprünglich direkt im PicATU500 waren, sind jetzt in die Fernbedienung ausgelagert.

1.1.7.1 Richtkoppler, Abstimmhilfe

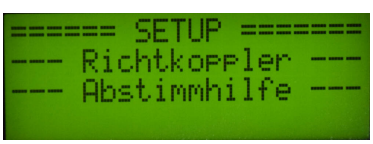
Mit dieser Funktion lässt sich der Richtkoppler sehr schön auf das Maximum der Richtschärfe abgleichen.

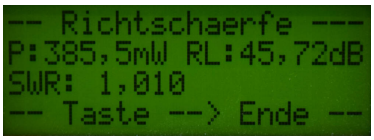


Mit dem Einstellregler wird die maximale Richtschärfe bei niedriger Frequenz eingestellt. Mit dem Trimmer wird anschließend die maximale Richtschärfe bei sehr hoher Frequenz eingestellt. Dazu muss aber der Deckel aufgeschraubt werden.



Der Abschlusswiderstand 50Ohm ist von hoher Qualität.

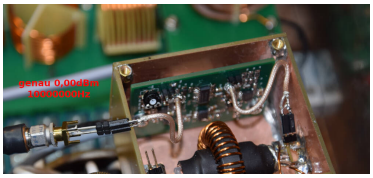
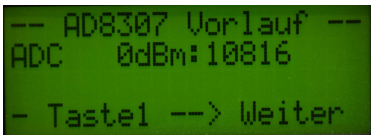




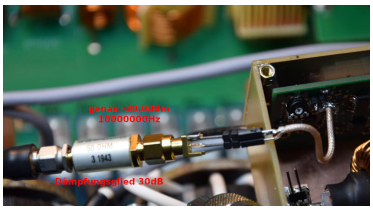
Zuerst bei 3,6MHz ReturnLoss 50dB, dann bei 28,5MHz ReturnLoss 39dB, zur Kontrolle bei 10MHz ReturnLoss 46dB. Diese Werte sind ausgezeichnet. Mit der Taste beenden wir diese Funktion.

1.1.7.2 Kal.AD8307 vor, mit HF Generator

Wir starten die Funktion „Kalibrieren AD8307 vorlauf“.

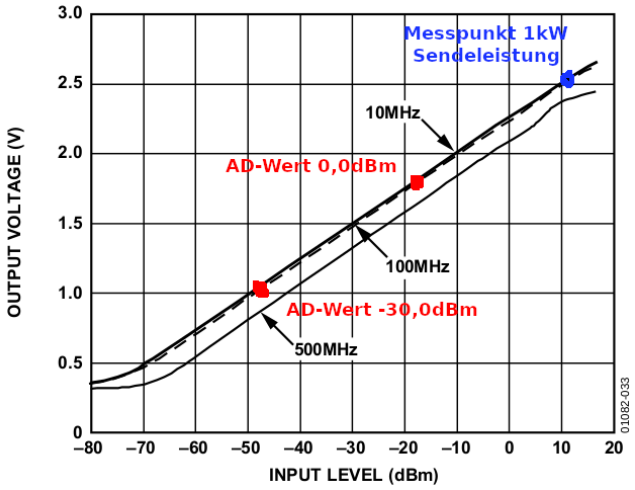
Die Verbindung zum Übertrager trennen wir ab und mit einem genauen Kalibrierpegel von 0,0dBm ermitteln wir den ersten Wert des A/D Wandlers im PIC. Die Zahl ist sehr groß die Wandlerbreite im PIC ist nur 10Bit (max 1023). Ich addiere aber bei jedem Messdurchgang 32 Einzelmessungen. Daher die große Zahl „10816“.




Die zweite Messung machen wir mit einem geringeren Pegel von -30dBm. Dazu wird ein gutes Dämpfungsglied vor dem Messeingang geschraubt. Die zweite Zahl des A/D-Wandlers beträgt „5951“.

Aus diesen beiden Zahlen des A/D-Wandlers werden die *Funktionsvariablen kx und ky* berechnet. *kx* ist der Anstieg der linearen Funktion und *ky* ist die Verschiebung auf der Y-Achse.

Die Berechnungen der Funktion AD-Wert zu dBm



Hier das Diagramm aus dem Datenblatt des AD8307. Ich habe die 2 Messpunkte als roten Punkt eingezeichnet. Sie befinden sich im Linearen Bereich der Funktion Input-dBm zu Output-Spannung.

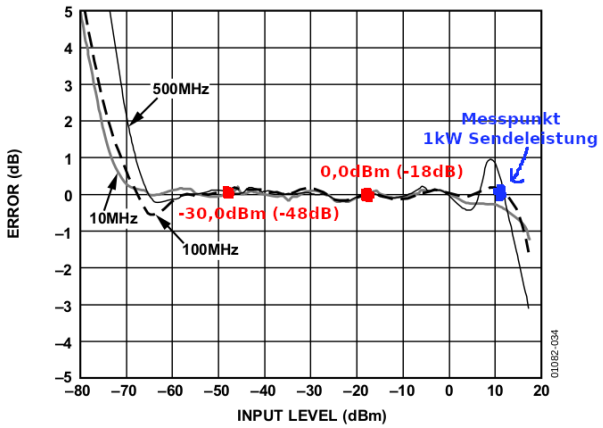


Figure 34. Logarithmic Law Conformance at 10 MHz, 100 MHz, and 500 MHz

Das ist das Error-Diagramm aus dem Datenblatt. In diesem Diagramm kann man sehen in welchem Bereich der AD8307 noch sauber arbeitet. Selbst bei 1kW Sendeleistung befinden wir uns mit der Messung noch im linearen Bereich. Ich habe den Messpunkt für 1kW in beiden Diagrammen eingezeichnet.

1kW Sendeleistung entspricht +60dBm. Wenn wir die Auskoppeldämpfung des Richtkopplers und die Dämpfung des Netztwerkes vor dem AD8307 abziehen, kommen wir auf die +11dBm die ich eingezeichnet habe.

$$+11dBm = +60dBm - 31dB - 18dB$$

Ihr wundert euch bestimmt, dass der 0,0dBm Messpunkt bei -18dBm liegt. Im Schaltbild der Messplatine sehen wird vor dem AD8307 ein Netzwerk mit Widerständen und einem Kondensator. Diese Netzwerk hat eine Dämpfung von etwa 18dB im gesammten Kurzwellenbereich. Ich habe in früheren Beschreibungen die Messpunkte falsch eingezeichnet. Ein OM aus Thüringen hat mich

darauf aufmerksam gemacht. Ich habe die Messpunkte jetzt korrigiert. **Durch meine Kalibrier-Berechnung geht das gesamte Netztwerk mit in die Kalibrierung ein.** Die Auskoppeldämpfung des Richtkopplers darf ich bei den roten Messpunkten nicht mit einrechnen. Der HF-Generator wird ja direkt an die Messplatine angeschlossen.

Nun zu den Berechnungen:

Bezeichner	Erklärung
dBmpegel1	Messpegel1 in dBm
dBmpegel2	Messpegel2 in dBm
adcwertpunkt1	Werte des AD Wandlers im PIC beim Pegel 1
adcwertpunkt2	Werte des AD Wandlers im PIC beim Pegel 2
mkx	Wert X, Anstieg der linearen Funktion
mky	Wert Y, Verschiebung Y-Achse der linearen Funktion

Formel 1: $mkx = \frac{dbmpegel1 - dbmpegel2}{adcwertpunkt1 - adcwertpunkt2}$

Formel 2: $mky = (adcwertpunkt1 * mkx * -1) + dbmpegel1$

Da der erste Pegel 0dBm beträgt, vereinfacht sich die Formel etwas.

Formel 1: $mkx = \frac{30dBm}{adcwertpunkt1 - adcwertpunkt2}$

Formel 2: $mky = adcwertpunkt1 * mkx * -1$

$$mkx = \frac{30dBm}{10816 - 5951} = 0,006166495375$$

$$mky = 10816 * 0,006166495375 * -1 = -66,6968$$

Wir habe jetzt die beiden Variablen für die Berechnung der *dBm-Werte* aus dem Wert des A/D-Wandlers.

Mit diesen beiden Werten aus der Berechnung der Kalibrierung können wir aus der Integer-Zahl des *ADC-Wertes* den Pegel in dBm berechnen. Die Formel für die Berechnung des HF-Pegels in *dBm* ist einfach und lautet:

$$Pegel(dBm) = ADCWert * mkx + mky$$

Zum Test setzen wir mal den A/D-Wert der -30dBm-Messung ein. Das muss wieder -30dBm ergeben.

$$Pegel(dBm) = 5951 * 0,006166495375 + -66,6968 = -29,99998826dBm$$

und jetzt noch den A/D-Wert der 0dBm-Messung. Das muss 0dBm ergeben.

$$Pegel(dBm) = 10816 * 0,006166495375 + -66,6968 = 0,00000992dBm$$

Die Formel funktioniert wie die Test-Berechnungen zeigen.

In die Berechnung der Leistung **mit Richtkoppler** muss letztendlich die Aus-

koppeldämpfung des Richtkopplers mit einfließen. Deshalb ergänze ich die Formel noch um die Auskoppledämpfung. Die Auskoppeldämpfung errechnet sich einfach aus der Windungszahl der Ringkernübertrager:

$$RK - Loss(dB) = log(Windungen) * 20dB$$

$$RK - Loss(dB) = log(36) * 20dB = 31,126dB$$

Jetzt können wir die Berechnung der Leistung aus dem ADC-Wert vervollständigen:

$$Pegel(dBm) = ADCWert * mkx + mky + 31,126dB$$

Ein Rechenbeispiel folgt noch. Der ADC-Wandler „Vorlauf“ ermittelt einen Wert von 15000. Das setzen wir in die Formel ein:

$$15000 * 0,006166495375 + -66,6968 + 31,126dB = 56,93dBm$$

Umgerechnet in Watt:

$$10^{\frac{56,93dBm}{10dB}} = 492791mWatt = 493Watt$$

Der ADC-Wert von „15000“ entspricht also einer gemessenen Vorlaufleistung von *493 Watt*.

1.1.7.3 Kal.AD8307 rück, mit HF Generator

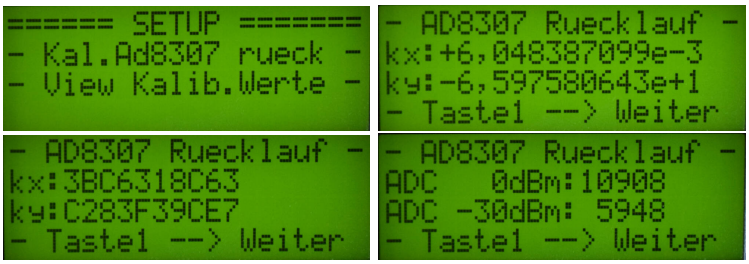
Für den *AD8307-Rücklauf* werden auch wieder mit dem HF-Generator und zwei Pegel 0dBm und -30dBm der Verlauf der linearen Funktion ermittelt. Da auf der Messplatine die HW nicht ganz identisch mit der HW Vorlauf-Messung ist ergeben sich etwas andere Zahlenwerte von „mkx“ und „mky“. Die Formeln für die Berechnungen sind die gleichen wie bei der Kalibrierung des „AD8307-Vorlauf“.

1.1.7.4 Kal.AD8307 vor, View Kalib.Werte

Mit dieser Funktion können wir uns alle Werte noch einmal anschauen.

1.1.7.5 Kal.AD8307 rück, View Kalib.Werte

Mit dieser Funktion können wir uns alle Werte noch einmal anschauen.



Sogar die A/D-Wandler-Werte berechnet die Funktion rückwärts aus kx und ky.

1.1.7.6 Kal.AD8307 vor, manuell ADC-Werte

Es werden die beiden Messwerte eingetragen und schon ist man fertig.

Möchte man die Kalibierwert kx und ky erneut berechnen braucht man nur die beiden Zahlen der A/D-Wandler-Werte eingeben und ist fertig, auch ohne genauen HF-Generator.

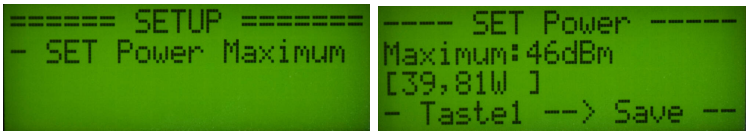
1.1.7.7 Kal.AD8307 rück, manuell ADC-Werte

Das gleiche gilt auch für den zweiten AD8307 für die Messung des Rücklaufes. Möchte man die Kalibierwert kx und ky erneut berechnen braucht man nur die beiden Zahlen der A/D-Wandler-Werte eingeben und ist fertig, auch ohne genauen HF-Generator.

1.1.7.8 SET Power Minimum

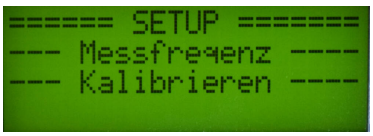
Hier legen wir fest ab welchem Pegel der Tuner beginnen soll zu tunen. Aber nur wenn wir das Tunen über einen Menüpunkt starten. Den Default-Pegel habe ich auf 24dBm gesetzt. Das entspricht 250mW.

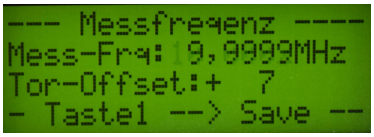
1.1.7.9 SET Power Maximum



Wichtig ist auch das Pegel-Maximum festzulegen. Ist der Pegel größer als das Pegel-Maximum beginnt der Tuner auch nicht mit dem Tunen. Das dient dem Schutz der Relais. Bei großer Leistung würden sonst die Relaiskontakte verbrennen. Den Default-Pegel habe ich auf 46dBm gesetzt. Das entspricht 40W Sendeleistung.

1.1.7.10 Messfreqenz, Kalibrieren

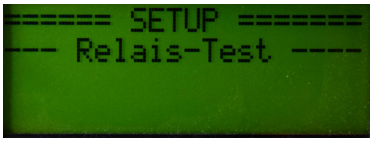




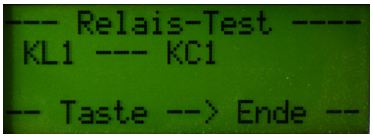
Mit dem Verstellen des Tor-Offset versuchen wird die gemessene Frequenz der tatsächlichen angelegten Frequenz zu nähern. Ich habe am Messeingang-Vorlauf der Messplatine einen Pegel von 0dBm und genau 10,000000MHz angelegt. Der Pegel kommt aus meinem Frequenz-Pegel-Normal, was ich mir einmal aufgebaut habe.

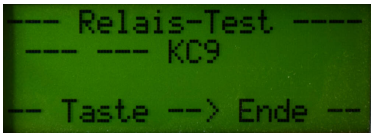
Wird der Tuner mit eigener Frequenzmessung betrieben, sollte die gemessene Frequenz auch stimmen. Das die Mess-Tor-Zeit nicht sehr groß ist und der Quarz am PIC auch etwas in der Frequenz abweicht habe ich diese Funktion implementiert.

1.1.7.11 Relais-Test

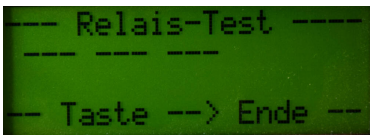


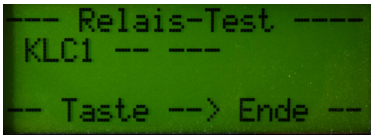
Hier starten wir auch noch mal den Relaistest. Die Relais müssen der Reihe nach getestet werden. Es sollte immer das/die Relais anziehen wie im Display geschrieben steht.





An Beginn ziehen beide Relais L-Glied und C-Glied KL1 bis KL8 und KC1 bis KC8. Rechts sehen wir das es einzeln mit dem C-Glied weiter geht KC9 bis KC11.





Dann folgt alle Relais abgefallen und weiter geht es mit den Relais für die Umschaltung der LC-Varianten, Relais KLC1 bis KLC4.

Hie noch einmal die Reihenfolge wie die Relais schalten müssen.

1. KL1 und KC1 EIN
2. KL2 und KC2 EIN
3. KL3 und KC3 EIN
4. KL4 und KC4 EIN

5. KL5 und KC5 EIN
6. KL6 und KC6 EIN
7. KL7 und KC7 EIN
8. KC8 EIN
9. KC9 EIN
10. KC10 EIN
11. KC11 EIN
12. alle Relais AUS
13. KLC1 EIN
14. KLC2 EIN
15. KLC3 EIN
16. KLC4 EIN

Kapitel 2

Erste Antennen-Anpassversuche

Funktioniert die Datenübertragung zwischen der Fernsteuerung und des PicATU500, können wir die Antenne und den TRX an den PicATU500 anschließen. Die Sendeleistung sollte gering sein. Es reichen schon 0,5 Watt.

An der Fernbedienung „Match“ starten Innerhalb von 30 Sekunden sollte das Sendesignal am PicATU500 anliegen. Sind die 30 Sekunden verstrichen ohne zu senden, geht der PicATU500 wieder in den Ruhezustand.

ein SWR < 1,5 wurde gefunden Ist das SWR < 1,5 wird die gefundene Einstellung im 10kHz-Segment gespeichert. Ich mache es bei einer neuen Antenne immer so, dass ich mit dem Befehl „Band save“ die gefundene Einstellung für alle 10kHz-Segmente im Band ablege. So habe ich eine Einstellung für jedes 10kHz-Segment im Band und brauche mit dem Befehl „ReMatch“ einfach nur Nachstimmen.

keine Abstimmung gefunden Beim Befehl „Match“ wird nach der ersten Suche geprüft ob das SWR < 2,5 ist. Wird diese Grenze nicht erreicht schaltet die LC-Variante um und der Match beginnt von vorn. Wird in allen LC-Varianten nichts gefunden, sollte man mit „Match deep“ noch einmal von vorn beginnen. Jetzt ist die Grenze SWR 2,5 aufgehoben. In jeder LC-Variante wird bis zum Ende gesucht. Es wird fast immer eine Anpassung gefunden.

immer noch keinen Erfolg In besonders hartnäckigen Fällen kann man im Menü mit „LC-Variante change“ die LC-Variante per Hand wechseln und anschließend mit dem Befehl „ReMatch deep“ ausführlich suchen. So hat man die Möglichkeit auch mit der LC-Variante *nur L* oder *nur C* eine neuen Anpassversuch mit „ReMatch deep“ zu starten.

„LC-Variante change“ lohnt sich auch wenn man mit dem SWR nicht ganz zufrieden ist. L und C untereinander vertauschen und noch einmal „ReMatch deep“ starten. Manchmal wird das SWR dadurch verbessert.

Einstellung speichern Ich hatte oben schon erwähnt, dass mit „Band save“ die gefundene Match-Einstellung in allen 10kHz-Segmente des Bandes ge-

speichert wird. Eingeschränktes Speichern für einen bestimmten Frequenzbereich geht besser mit „10kHz save +/− #0kHz“ oder „10kHz save −a +b“.

Ich wünsche viel Erfolg beim Anpassen der Antenne.

Kapitel 3

Schlusswort

Dieses Projekt darf nicht kommerziell vermarktet oder genutzt werden. Alle Rechte liegen bei DL4JAL (Andreas Lindenau). Ich wünsche viel Spaß beim Basteln.

vy 73 Andreas DL4JAL

✉ DL4JAL@t-online.de
☞ www.dl4jal.de