PicATU500 (SymTuner 750W)
Fernsteuerung-Baugruppe
SW-Version 1.14

Andreas Lindenau DL4JAL

22. November 2022

Zusammenfassung

Der „PicATU500“ ist ein Bastelprojekt von mir. Nachdem die kleineren PicATU100 und PicATU20 so gut funktionieren, habe ich mich entschlossen noch eine 500 Watt-Version zu konstruieren. Die FW im PIC habe ich entsprechend angepasst. Das Schreiben der Match-Funktionen hat sich über mehrere Jahre hingezogen. Jetzt funktioniert die automatische Anpassung fast perfekt. Als Programmiersprache verwende ich Assembler.

Ab der Firmware-Version 1.06 ist es möglich meinen alten SymTuner 750W mit Schrittmotor auch mit dieser Fernsteuerung zu steuern. Die Erkennung erfolgt automatisch. Voraussetzung: der SymTuner 750W wurde umgebaut auf die neue Fernsteuerung mit RS232-Stromschleife.

Für die Fernsteuerung des PicATU500 habe ich eine extra Baugruppe entwickelt.

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.

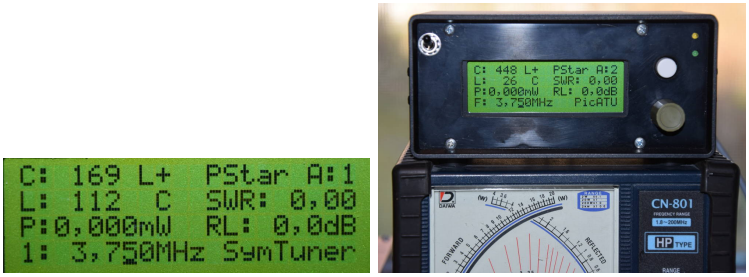
1.1 Software/Firmware

Grundsätzlich wird der Tuner über die Fernsteuerung betrieben!

Die Software wurde alles in Assembler geschrieben. In der Fernsteuer-BG habe ich einen PIC18F26K22 eingesetzt. Es ist auch möglich den PIC18F25K22 zu verwenden. Die Taktfrequenz beträgt 18,432MHz und wird mit einem Quarz erzeugt. Den Quarz habe ich wegen der Erzeugung der genauen Baudrate eingesetzt. Der PIC18F26K22/25K22 hat zwei RS232 Schnittstellen. Die erste RS232 benutze ich für die Verbindung zum PicATU500/SymTuner. Mit der zweiten RS232 wird die CAT-Verbindung zum Transceiver hergestellt. Eine 3. RS232 habe ich nachgebildet (nur TTL-Pegel). Damit wird mein Split-Filter (Frequenz-Info) gesteuert, so dass immer der richtige Tiefpass/Hochpass aktiv wird.

1.1.1 Automatische HW-Erkennung des Tuner-Typ

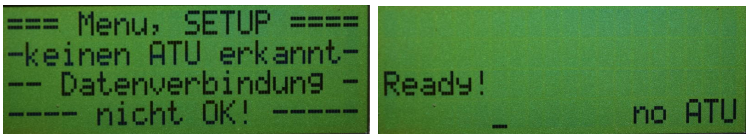
Neu ab der Version 1.06 der Firmware ist die HW-Erkennung des Tuners. Mit dieser Fernsteuerung ist es möglich auch meinen älteren sogenannten *SymTuner 750W mit Schrittmotor und RS232 Stromschleife* fernzusteuern.



Links, der alte *SymTuner 750W mit Schrittmotor* wurde erkannt. Rechts, *PicATU500*.

Nach Power-ON erfolgt eine Datensatz-Anforderung an den Tuner. Sendet der PicATU500 den Antwort-Datensatz steht in Zeile 4 rechts „PicATU“. Sendet der ältere SymTuner den Antwort-Datensatz steht in Zeile 4 rechts „SymTuner“. 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)**

Keinen Tuner erkannt, SETUP 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 1.1.5.1 auf Seite 15.

Firmware Versionen? Aufruf der FW-Version.

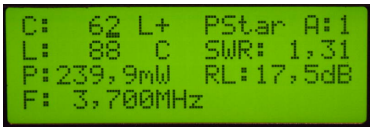
TRX-CAT Monitor, Hexadezimal Mit dem Monitor wird sichtbar wenn Infos über die CAT-Schnittstelle kommen. Alle Zeichen in HEX-Darstellung. Siehe auch Kapitel 1.1.5.7 auf Seite 18.

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 1.1.5.8 auf Seite 18.

Break!! Abbruch des SETUP.

1.1.2 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**.

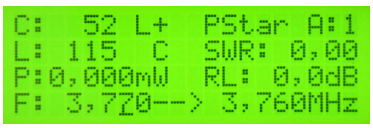
Taste im Drehgeber *kurzer Tastendruck* Der Kursor rückt um eine Stelle weiter. Je nachdem welcher Modus (Modus:Frequenzänderung oder Modus:LC-Änderung) aktiv ist.

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

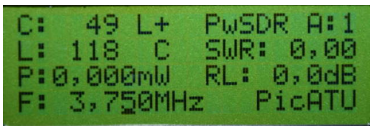
Nach einem *langen Drehgebertastendruck* wechselt der Kursor zum C-Wert/Einerstelle. Mit *kurzen Drehgebertastendruck* schaltet der Kursor weiter zur nächsten Stelle. Mit dem Drehgeber können wir die LC-Werte verstellen. Jede Änderung wird sofort zum ATU gesendet.

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

1.1.3 Frequenzinformation und CAT-Anbindung des TRX

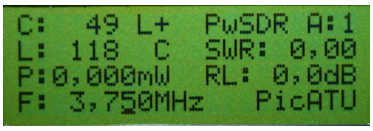
Soll die Tuner-Abstimmung aus dem Speicher des PicATU500 ausgelesen werden braucht der PicATU500 eine Frequenzinformation. Ich liste mal die verschiedenen Möglichkeiten der Frequenzinformation auf:

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“.

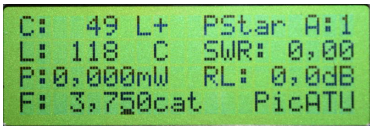
Frequenzmessung RK. oder Fernbedienung ohne TRX-Cat Die Frequenz kann auch an der Fernbedienung mit dem Drehgeber eingestellt werden. Siehe Kapitel 1.1.2. Die *Frequenzmessung im Richtkoppler* wird aber weiterhin benutzt für das Auslesen der Einstellungen aus dem Speicher. Sichtbar im Display Fernsteuerung 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.

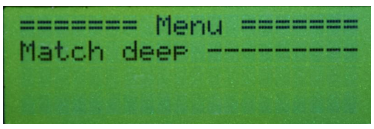


Die Anzeige der Frequenz ändert sich in der Fernbedienung „MHz“ wird durch „cat“ ersetzt und im PicATU500 ändert sich der Buchstabe nach der Frequenz von „M“ zu „c“.

1.1.4 Menübefehle, bei HW-Erkennung PicATU500

Als erstes beschreibe ich die Menübefehle, wenn als Tuner der *PicATU500* erkannt wurde (*das ist der Normalfall*). 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 1.1.5.6 auf Seite 18, wenn sie aktiv ist.**

1.1.4.1 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 1.1.4.2 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) und die Wertigkeit von C von 0 (0,0pF) bis 1024 (1600pF). 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).

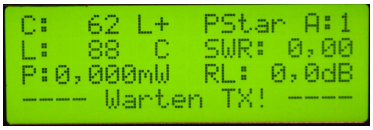
Wie schon oben kurz erwähnt suchen die Funktionen *Match* und *Match deep* bis das SWR den Wert 1,2 unterschreitet. Dann bricht die Funktion ab mit „Abstimmung OK“. Mit *ReMatch* kann man anschließend die Suche fortsetzen und ein noch besseres SWR finden.

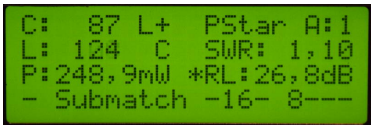
1.1.4.2 ReMatch

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.

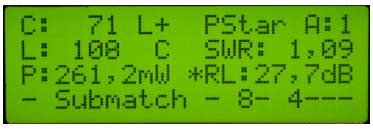
Beispielbilder von „ReMatch“ Hier noch ein Beispiel einer Nachabstimmung. Mit allen Zwischen-Display-Bildern.

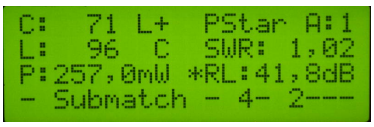
Zuerst die Kontrolle wie das SWR vor *ReMatch* war. Im Menu den Befehl *ReMatch* starten.



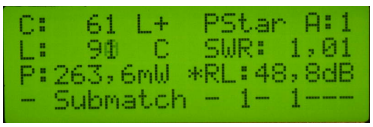


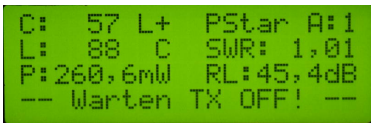
Der PicATU500 wartet auf das Sendesignal. Die Suche beginnt in einem größeren LC-Feld. Das erste Suchergebnis *SubMatch mit Schrittweite L und C* wird im Display angezeigt.



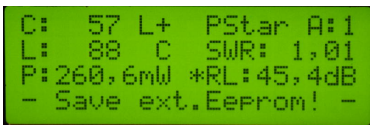


Nach jeden *SubMatch* wird ein Zwischen-Datensatz übermittelt. Das Display wird aktualisiert. Das SWR wird immer besser und die Schrittweiten kleiner.



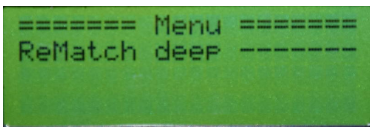


Ein gutes SWR. Ein super Ergebnis. Die Schrittweite ist beim kleinsten Wert angekommen. Anschließend wird mit großer Messzwischenzeit noch einmal im MatrixLC 4x4 nachgestimmt. Return Loss ändert sich noch etwas. Am „Match ENDE“ wartet der Tuner bis der Sender AUS ist.



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

1.1.4.3 ReMatch deep

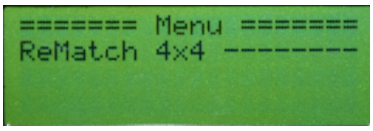


Die Funktion *ReMatch deep* sucht eine Anpassung in der eingestellten LC-Variante von Anfang an mit Grundmatch, Submatch usw., wie in der Funktion *Match* beschrieben. Diese Funktion habe ich zusätzlich programmiert für die ganz „hartnäckigen Fälle“. *ReMatch deep* ist die *Match deep* Funktion ohne 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.

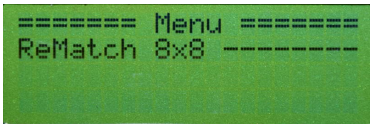
Der Abstimmvorgang geht so schneller, teilweise erheblich schneller.

1.1.4.4 ReMatch 4x4



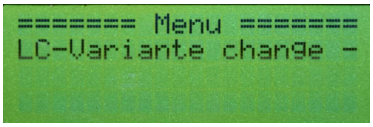
Von den momentanen L- und C-Werten aus gehend werden +/− 2 Werte auf der X-Achse angenommen und +/− 2 Werte auf der Y-Achse. Es ergeben sich (4x4) 16 Suchfelder die der Reihe nach eingestellt werden und das *Return Loss* verglichen wird. Das Suchfeld mit dem höchsten *Return Loss* wird anschließend eingestellt. 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).

1.1.4.5 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).

1.1.4.6 LC-Variante



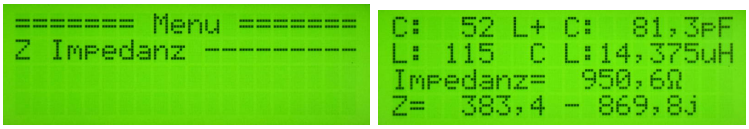
Diese Funktion kann die L/C Variante gezielt ändern. Im LCD-Display werden die LC-Varianten angezeigt. Die entsprechende LC-Variante wählen und mit der Einzeltaste bestätigen. Die Funktion *LC-Variante* braucht man, wenn in der Variante *nur-L* und *nur-C* nach einer Impedanzanpassung gesucht werden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

1.1.4.7 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.

Es soll ja nur eine zusätzliche Information sein.

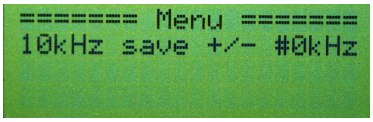


Zusätzlich zur Impedanz wird auch die Größe des LC-Gliedes (in uH und pF) mit angegeben und welche LC-Variante geschaltet ist.

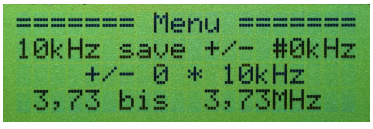
1.1.4.8 Band save

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.

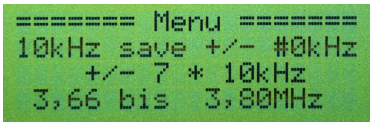
1.1.4.9 10kHz save +/− #0kHz



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.

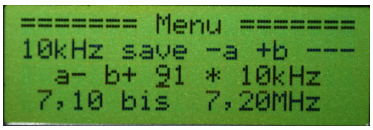


Noch ein Beispiel, Auswahl #7: die Frequenz beträgt 3,730 000 MHz: die Relaiseinstellung wird in $+/-$ 7 $*$ 10kHz Segment abgespeichert. So wie das im Display Zeile 4 steht.

1.1.4.10 10kHz save $-a+b$

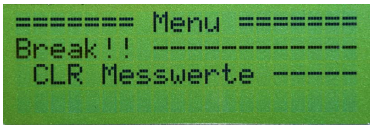
Diese Funktion ist ähnlich der Funktion im vorherigen Kapitel. Es wird die Relaiseinstellung der aktuellen Frequenz im Segment 10kHz gespeichert. Zusätzlich kann in den Nachbarsegmenten (bis zu 9 Segmente) die Einstellung mit gespeichert werden. In dieser Funktion kann man eine getrennte Einstellung 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.

1.1.4.11 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.

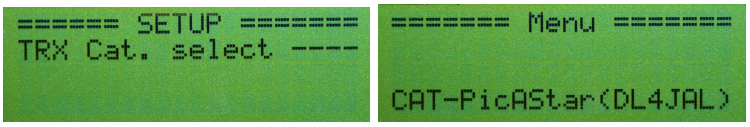
1.1.5 SETUP, bei HW-Erkennung PicATU500

Als erstes beschreibe ich die SETUP-Befehle, wenn als Tuner der *PicATU500* 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.

1.1.5.1 TRX Cat. select

Diese Funktion wählt den Typ des TRX aus.



An der Fernsteuer-Baugruppe ist noch eine 2. Schnittstelle für den Anschluss des TRX-CAT-Verbindung vorhanden. Die Schnittstellen müssen zum HW-Protokoll des TRX passen.

Folgende HW-Anschlüsse habe ich vorgesehen:

J6, 5V RX-TTL Diesen Anschluss nutze ich für meinen PicAStar. Aber auch die Icom Transceiver mit einem CI-V-Anschluss.

J5, RS232 Eine normale RS232 mit einer zusätzlichen kleinen Platine mit RS232-IC (z.B.:MAX232).

J7, USB extern Auf der MC-Leiterplatte ist ein IC FT232RL aufgelötet für eine USB-Verbindung zu PC. Die USB-Buchs muss extra montiert werden.

Die 3 Stecker J5, J6 und J7 arbeiten parallel. Es darf jeweils nur eine Möglichkeit benutzt werden, sonst gibt es „Datensalat“.

In der Software habe ich folgende TRX berücksichtigt:

CAT-PicAStar(DL4JAL), J6 Der CAT-Anschluss ist ein Ausgang mit 5V TTL-Pegel. Bei jedem VFO-Frequenzwechsel wird sofort die neue Frequenz ausgegeben.

CAT-FT847, J5 Den FT847 habe ich auch angepasst. Das ist aber schon einige Jahre her. Ob das noch zuverlässig funktioniert weiß ich nicht.

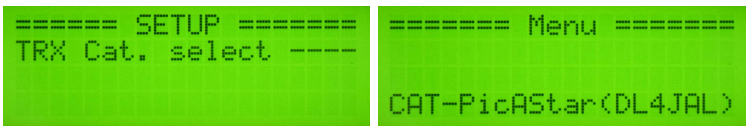
CAT-Icom CI-V Remote, J6 Der Icom TRX funktioniert auch an der Fernsteuerung.

CAT-K2/K3 Elecraft, J5 Funktioniert erst ab FW 1.13 richtig. Baudrate ist fest eingestellt 4800 Baud.

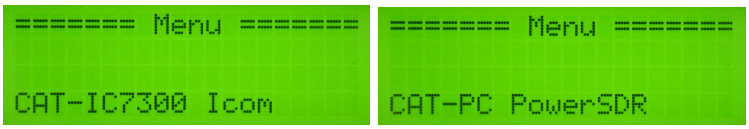
CAT-PC PowerSDR, J7 Die Windows-Software „PowerSDR“ funktioniert auch mit der CAT-Anbindung. Entweder über die einfache RS232 oder über USB.

CAT-FTDX101 Yaesu, J5 Über die RS232. FTDX Einstellung 9600Baud.

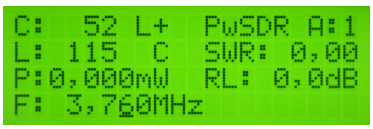
Bei Elecraft ist die default-Baudrate 4800 Baud. Alle anderen TRX arbeiten mit 9600 Baud.



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



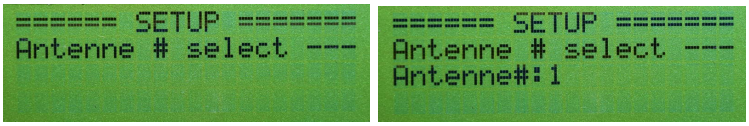
Links der Icom CI-V und rechts die PC-Software PowerSDR.



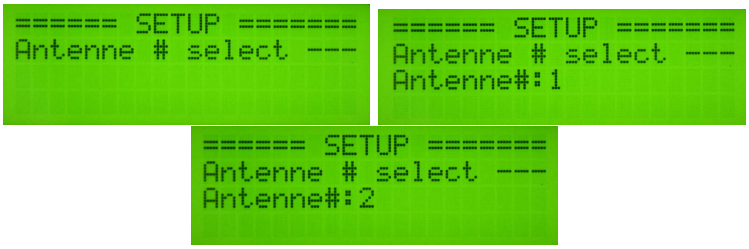
Ich habe gewechselt zur *PC-Software PowerSDR*. Siehe Zeile 1 rechts.

Die Auswahl wird mit der Einzeltaste bestätigt. Ein *langer Tastendruck* führt zum Abbruch der Funktion.

1.1.5.2 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).

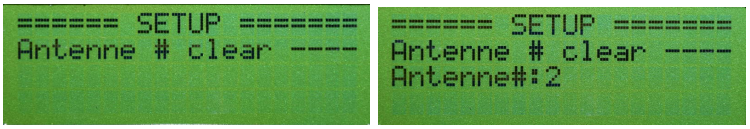
Antenne#	Adressbereich	HEX
1	0 bis 11400	0 – 0x2C88
2	11401 bis 22800	0x2C89 – 0x5910
3	22801 bis 34200	0x5911 – 0x8598
4	34201 bis 45600	0x8599 – 0xB220
5	45601 bis 57000	0xB221 – 0xDEA8

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

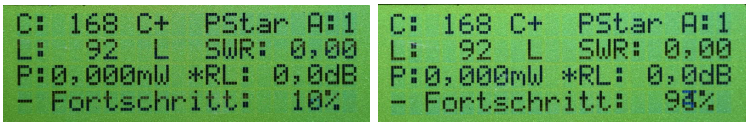
Welche Antennen-Nummer (Speicherbereich) aktiv werden soll wird in dieser Funktion festgelegt. Verschiedene Antennen-Nummern sind vorteilhaft beim Testen anderer Antennen. So kann man die einmal gefundenen Einstellungen und Werte erhalten. Möchte man eine neue Antenne testen, braucht man nur auf eine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

1.1.5.3 Antenne # clear



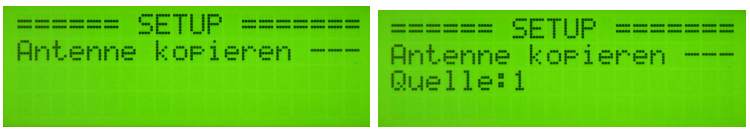
Die Funktion *Antenne # clear* löscht alle Speicherzellen für den Adressbereich der entsprechenden Antennen-Nummer. Ein Rückmeldung in Prozent erfolgt über die Fernsteuerung während des Löschvorgangs. Das Löschen dauert etwas länger. Ein *langer Tastendruck* führt zum Abbruch der Funktion.



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

1.1.5.4 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.

1.1.5.5 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.

1.1.5.6 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.

1.1.5.7 TRX-CAT Monitor, Hexadezimal

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

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

Taste 2, Taste im Drehgeber LCD löschen. Beim FTDX101 zusätzlich VFO-Daten anfordern, Befehl *FA;* zum TRX. Beim K2/K3 Befehl *AI2;* zum TRX übertragen.

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.

1.1.5.8 TRX-CAT Monitor, Ascci

Dieser Monitor überwacht die RS232 zum TRX und schreibt jedes empfangene Byte als Ascci-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.

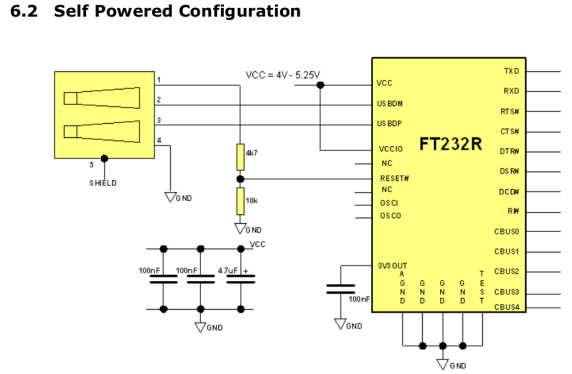
Taste 1, Einzeltaste Monitor beenden. Die FW startet neu.

Taste 2, Taste im Drehgeber LCD löschen. Beim FTDX101 zusätzlich VFO-Daten anfordern, Befehl *FA;* zum TRX. Beim K2/K3 Befehl *AI2;* zum TRX übertragen.

1.1.6 PowerSDR Schnittstelleneinrichtung (PC-Software)

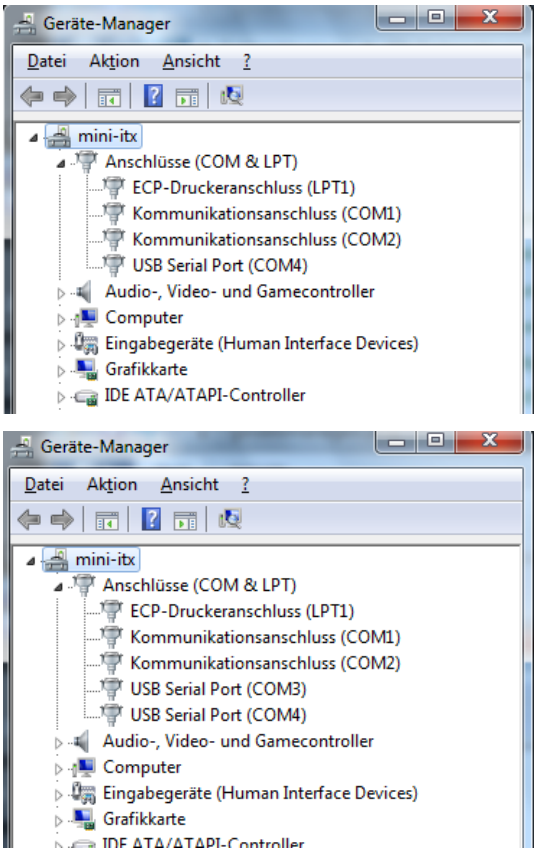
Zuerst stecken wir die Verbindung mit USB-Kabel „Remote-BG“ und „PC“. Im PC wird eine neue serielle Schnittstelle sichtbar. Jetzt machen wir im „PowerSDR“ folgende Einstellungen. Im „SETUP, CAT Control“ wählen wir die neue Schnittstelle aus (9600,none,8,1). Es folgt noch **ID as** „PowerSDR“ und bei „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.

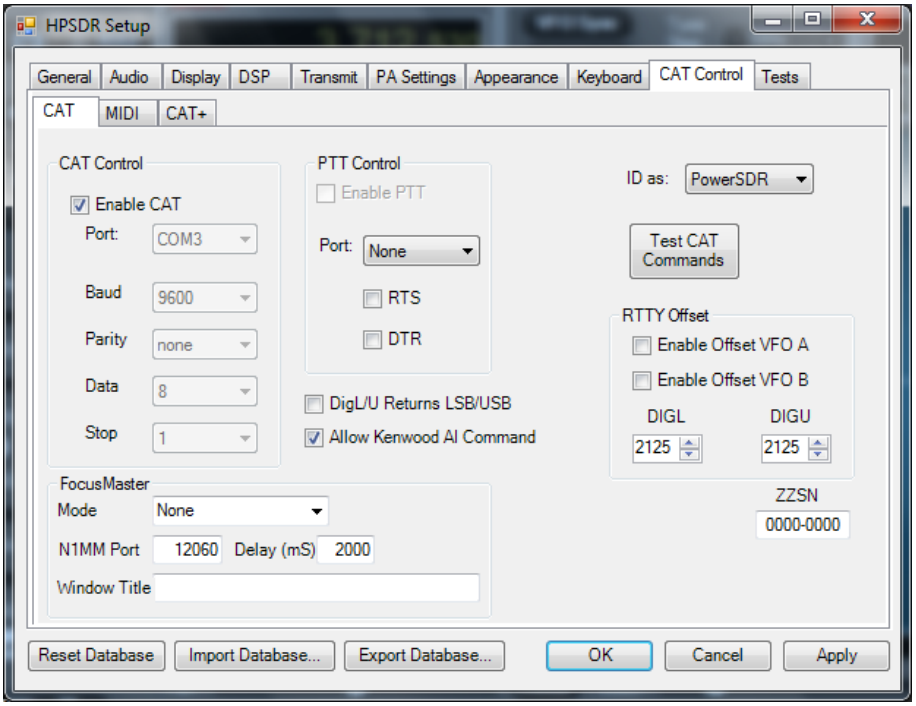


Die Beschaltung des FT232RL, wenn die Versorgungsspannung Baugruppe und 5 Volt USB getrennt sind. Der FT232RL wird beim anstecken des USB-Kabels neu gestartet (Reset).

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.



Wir sehen die neu entstandene COM. In diesem Fall die COM3.



Jetzt können wir im „PowerSDR“ die neu entstandene Schnittstelle benutzen.
Hier die „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.

1.1.7 Menübefehle, bei HW-Erkennung SymTuner 750W

Es folgen die Menübefehle, wenn als Tuner der *SymTuner 750W* erkannt wurde. Die Menü-Struktur ist eine andere als beim *PicATU500*.

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 sie mit der Einzeltaste.

1.1.7.1 Match ———-[47]

Der Match-Befehl ist so ähnlich wie in Kapitel 1.1.4.1 auf Seite 7 beschrieben. Nur das beim *SymTuner 750W mit Schrittmotor* kein durchgängiger C-Wertebereich möglich ist. Deshalb ist der Abstimmbereich immer nur so groß wie dem Drehkondensator sein Drehbereich (180°). Der Schrittmotor legt bei einer Drehung von 180°100 Schritte zurück. Für eine Kapazitätserhöhung werden Festkondensatoren zugeschaltet.

Wertebereich	Kapazität	Kapazität pro Schritt
0 bis 99	0 bis 37pF	0,37pF
100 bis 199	0 bis 140pF	1,4pF
200 bis 299	100 bis 240pF	1,4pF
300 bis 399	200 bis 340pF	1,4pF
400 bis 499	300 bis 440pF	1,4pF
500 bis 599	400 bis 540pF	1,4pF
600 bis 699	500 bis 640pF	1,4pF
700 bis 799	600 bis 740pF	1,4pF
800 bis 899	700 bis 840pF	1,4pF

Ein Beispiel: Der C-Wert ist 350. *Match* würde von 300 bis 399 und im vollen L-Bereich 0..255 eine Anpassung suchen. *Match* sucht also beim C-Wert nur innerhalb der 100 Schritte des Schrittmotors (300 bis 399) und nur innerhalb der eingestellten LC-Variante. Match beginnt immer mit *Grundmatch* (die Schrittweite steigt bei jedem Schritt quadratisch) und sucht anschließend mit *Submatch* (konstante Schrittweite) weiter. Bis das SWR kleiner 1,2 ist oder beide Schrittweiten 1 sind.

1.1.7.2 ReMatch neu —-[48]

ReMatch ist eine Nachstimmfunktion mit einem größeren Suchbereich, aber der Suchbereich ist nicht so groß wie bei *Match*, da nicht mit *Grundmatch* begonnen wird. Die Suche endet erst bei Schrittweite 1 von L und C und somit beim kleinsten SWR.

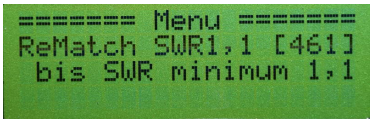
1.1.7.3 ReMatch alt —-[46]

Die alte Suchfunktion habe ich beibehalten. Sie dauert aber meistens sehr lange und folgt einer anderen Funktions-Logic.

1.1.7.4 ReMatch 8x8 —[460]

Das ist ein ganz einfaches Nachstimmen in einer Fläche von 8x8 vom L-Wert und C-Wert, Schrittweite 1. Das Feld mit dem höchsten Return Loss wird anschließend genommen.

1.1.7.5 ReMatch SWR1,1 [461]

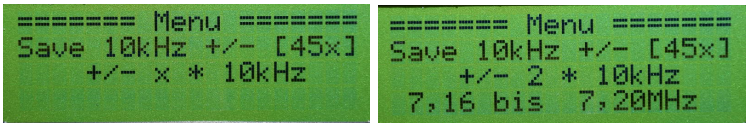


Die alte Suchfunktion [46] aber mit SWR-Grenze 1,1. Die Funktion geht schneller als *ReMatch alt [46]*, da bei einem SWR kleiner 1,1 sofort die Funktion beendet ist.

1.1.7.6 Save 10kHz ——[45]

Speichern der gefundenen Match-Einstellung in den Speicher. Die Speicherstelle richtet sich nach der Frequenz. Aller 10kHz ist ein neuer Speicherplatz. Frequenz 3,767 MHz ist Speicherplatz-Nummer 376.

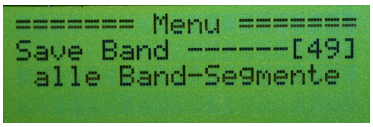
1.1.7.7 Save 10kHz +/- [45x]



Mit dem Drehgeber habe ich auf 2*10kHz eingestellt. Unten sehen wir für welchen Frequenzbereich die Einstellung gespeichert wird.

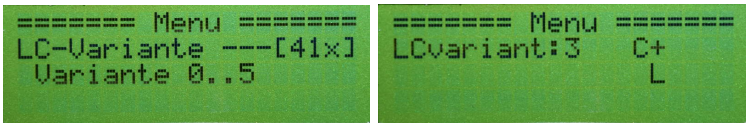
Mit dieser Funktion kann die gefundene Match-Einstellung in den Nachbar-10kHz-Segmenten mit abgespeichert werden. Mit dem Drehgeber wird der gewünschte Frequenzbereich eingestellt. Der maximal wählbare Frequenzbereich ist +/− 90kHz.

1.1.7.8 Save Band ——[49]



Die gefundene Match-Einstellung wird für den ganzen Frequenzbereich des Bandes im Speicher abgelegt und etwas darüber hinaus. Das erleichtert das Nachstimmen der Impedanzanpassung erheblich, da immer ein Ausgangswert für *ReMatch* im Speicher vorhanden ist.

1.1.7.9 LC-Variante —[41x]



Mit dem Drehgeber kann man die LC-Variante ändern. 6 verschiedene Varianten sind möglich. Mit dem Drehgeber wird die entsprechende Variante ausgewählt.

0 nur L in Reihe

1 L in Reihe C zur Masse, Impedanz größer 50 Ohm.

2 C zur Masse L in Reihe, Impedanz kleiner 50 Ohm.

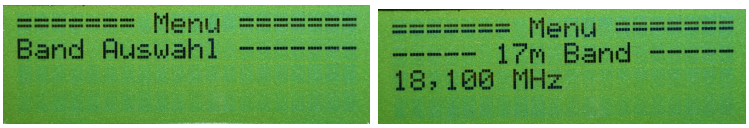
3 C in Reihe L zur Masse, Impedanz größer 50 Ohm.

4 L zur Masse C in Reihe, Impedanz kleiner 50 Ohm.

5 nur C in Reihe

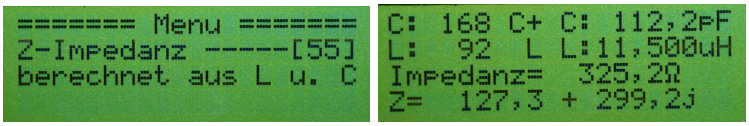
In der Praxis hat sich gezeigt das Vertauschen von L und C manchmal eine besser Anpassung ermöglicht.

1.1.7.10 Band Auswahl ——-



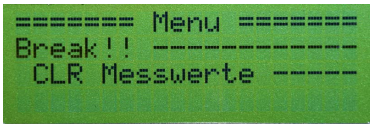
Diese Funktion habe zusätzlich mit geschrieben. Wird die Frequenz nicht per TRX-CAT eingestellt, kann mit dieser Funktion die Frequenz entsprechend der Bänder schnell umgeschaltet werden.

1.1.7.11 Z-Impedanz ——[55]



Die Berechnung der Impedanz habe ich in die Fernbedienung verlagert. Aus den L-Wert und C-Wert wird die Induktivität und die Kapazität berechnen. Anschließend wird mit der Frequenz und mit komplexer Berechnungen die Impedanz der Antenne ermittelt. Allerdings sind das nur Richtwerte, da die Impedanz-Abweichungen des Symmetrieübertragers SymBa800 bei hohen Frequenzen zu erheblichen Abweichungen bei den Berechnungen führen.

1.1.7.12 Break!! ———-



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.

1.1.8 SETUP, bei HW-Erkennung SymTuner 750W

Das SETUP ist wird durch die Erkennung des *SymTuners* auch verändert. Die ersten 5 Menüpunkte sind gleich. Sie verweisen auf vorherige Kapitel. Es kommen noch einige Funktionen hinzu.

1.1.8.1 TRX Cat. select

Siehe Kapitel 1.1.5.1 auf Seite 15.

1.1.8.2 Antenne # sel. [44x]

Siehe Kapitel ?? auf Seite ??.

1.1.8.3 Antenne # clear[51x]

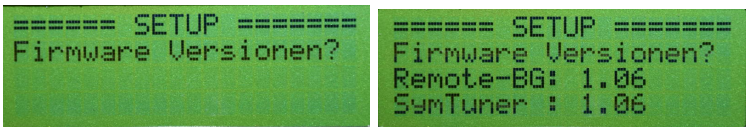
Siehe Kapitel 1.1.5.3 auf Seite 17.

1.1.8.4 Antenne kopier[52ab]

Siehe Kapitel 1.1.5.4 auf Seite 17.

1.1.8.5 Firmware Vers. [540]

Siehe Kapitel 1.1.5.5 auf Seite 18.



1.1.8.6 Frequenz-Erkenn[53x]

Der *PicATU* erkannt automatisch ob die Frequenz gemessen werden muss oder ob die Frequenzinformation vom TRX kommt. Dem *SymTuner* muss ich sagen was er machen soll.

1 Frequenzinformation nur per Remote. Übermittlung per CAT-Schnittstelle vom TRX und Weiterleitung zum Tuner.

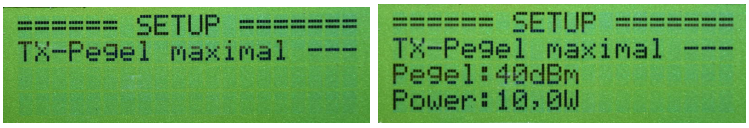
2 Frequenzerkennung nur durch Frequenzmessung in Tuner.

3 Frequenzerkennung gemischt entweder durch Frequenzmessung in Tuner oder per Remote-TRX.

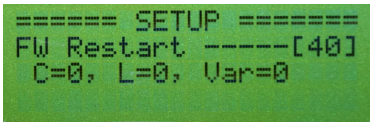
Es sollte bevorzugt mit der Einstellung „1“ gearbeitet werden. Die eingestellte Art wird im Eeprom gespeichert und ist nach PowerON wieder gültig.

1.1.8.7 TX-Pegel max. [56xx]

Der maximale Sendepegel in *dBm* wird hier festgelegt. Wird der Pegel überschritten schalten die Relais nicht mehr. Die Relaiskontakte sollen so geschützt werden. Nicht das aus Versehen bei 500W Sendeleistung der Match-Befehl gestartet wird.

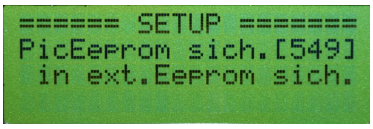


1.1.8.8 FW Restart ——[40]



Es wird ein Neustart der Firmware des PIC im Tuner aktiviert. Es ist das gleiche wie PowerON am Tuner. Der NULL-Punkt des C-Wertes wird gesucht (Schrittmotor) und alles wird auf NULL gesetzt.

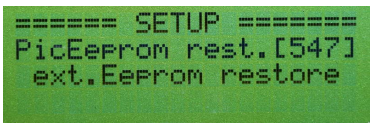
1.1.8.9 PicEeprom sich.[549]



Der PIC im Tuner hat seinen eigenen Eeprom-Bereich. Im Eeprom werden wichtige Daten gespeichert. Zum Beispiel die Kalibrierung des Messkopfes. Mit diesem Befehl kann der gesamte Eeprom-Inhalt im externen Eeprom (ab Adresse 0xDF00, 57088) gesichert werden. Das sind nur 255 Byte.

Das ist zum Beispiel vor einem Firmware-Update wichtig oder einem Wechsel des PIC im Tuner.

1.1.8.10 PicEeprom rest.[547]



Die Eeprom-Sicherung wird vom externen Eeprom zurück zum PIC kopiert. Anschließend wird ein Neustart der Firmware im Tuner ausgeführt.

1.1.8.11 TRX-CAT Monitor, Hexadezimal

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

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

Taste 2, Taste im Drehgeber LCD loeschen. Beim FTDX101 VFO-Daten anfordern.

Siehe auch Kapitel 1.1.5.7 auf Seite 18.

1.1.8.12 TRX-CAT Monitor, Ascci

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

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

Taste 2, Taste im Drehgeber LCD loeschen. Beim FTDX101 VFO-Daten anfordern.

Siehe auch Kapitel 1.1.5.8 auf Seite 18.

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 am Anfang immer so, dass ich mit dem Befehl „Band save“ die gefundene Einstellung für alle 10kHz-Segmente im Band ablege. So kann ich bei einer neuen Frequenz im Band 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 nicht 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 gespeichert wird. Eingeschränktes Speichern für einen bestimmten Frequenz-

bereich 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