

1. プログラム補正について

Arduino IDE をダウンロードした環境の PC が必要です。CQ 出版社の Web から SWR & POWER 計のプログラムをダウンロードし PC の USB ケーブルを Nano と接続し書き込みを行って下さい。同じ回路で作成してもタンデムマッチカブラ等の作成差や部品の差で検出値は異なります。

回路通りのハード部を作成していればプログラムをロードし電源を投入すれば動作します。

表示数値を計測し正しい値と合わせる必要が有ります。プログラム設定は左端の行数を確認して行ってください
筆者は送信機の出力計を基準に補正しましたが外部の POWER 計等使用でも同様です。

ケーブルを出来るだけ短くして接続してください。（送信機側の ATU は必ず OFF としてください）

1-1. 設定箇所-1 (LCD 電力表示について)

下記の①40 行～59 行目の②数値と±を調整する必要が有ります送信機と SWR & POWER 計、ダミーロードを最短距離のケーブルで接続してください。

7MHz（主調整は7MHzで行います）を入力します。POWER は【PTX】として表示します。この時の LCD 出力表示が送信出力と同じになるように差異を計算し変更してください。

例えば入力が 10W なのだが PTX 値が 9.5W の時、「+=0.50」と書き換えてください

100W 以上まで使用の場合、59 行と 60 行の間に電力幅の数値を変えて構文を追加してください

①

②

```
40 else if((forward_power > 1.11) && (forward_power <= 4.50)) forward_power -= 3.00;
41 else if((forward_power > 4.51) && (forward_power <= 9.50)) forward_power -= 4.20;
42 else if((forward_power > 9.51) && (forward_power <= 14.50)) forward_power += 4.20;
43 else if((forward_power > 14.51) && (forward_power <= 19.50)) forward_power += 4.10;
44 else if((forward_power > 19.51) && (forward_power <= 24.50)) forward_power += 4.00;
45 else if((forward_power > 24.51) && (forward_power <= 29.50)) forward_power += 4.70;
46 else if((forward_power > 29.51) && (forward_power <= 34.50)) forward_power += 5.40;
47 else if((forward_power > 34.51) && (forward_power <= 39.50)) forward_power += 5.10;
48 else if((forward_power > 39.51) && (forward_power <= 44.50)) forward_power += 4.80;
49 else if((forward_power > 44.51) && (forward_power <= 49.50)) forward_power += 3.80;
50 else if((forward_power > 49.51) && (forward_power <= 54.50)) forward_power += 4.60;
51 else if((forward_power > 54.51) && (forward_power <= 59.50)) forward_power += 3.60;
52 else if((forward_power > 59.51) && (forward_power <= 64.50)) forward_power += 6.60;
52 else if((forward_power > 64.51) && (forward_power <= 69.50)) forward_power += 6.60;
53 else if((forward_power > 69.51) && (forward_power <= 74.50)) forward_power += 5.60;
54 else if((forward_power > 74.51) && (forward_power <= 79.50)) forward_power += 4.50;
55 else if((forward_power > 79.51) && (forward_power <= 84.50)) forward_power += 3.40;
56 else if((forward_power > 84.51) && (forward_power <= 89.50)) forward_power += 3.00;
57 else if((forward_power > 89.51) && (forward_power <= 94.50)) forward_power += 0.60;
58 else if((forward_power > 94.51) && (forward_power <= 99.50)) forward_power += 1.55;
59 else if((forward_power > 99.51) && (forward_power <= 104.50)) forward_power -= 0.50;
60 return forward_power;
61
```

1-2. 設定箇所-2 (LCD BAND 毎の電力表示について)

BAND ごとの周波数偏差を補正します。下記の 65 行～74 行目のカッコ内の数値を調整する必要が有ります
1-1 において、7MHz で調整した値との偏差を補正します (最大電力 100W 時のみの補正となります)

調整は BAND ごとに行います。各 BAND で 100W を入力してください

この時各 BAND の PTX 値が送信出力と同じ 100W になるようにカッコ内の差異 (%) を計算し補正してください

65	#define ForwCal19MHz	(110.6)	// 1.8M band
66	#define ForwCal35MHz	(103.7)	// 3.5M band
67	#define ForwCal7MHz	(100.0)	// 7.0M band
68	#define ForwCal10MHz	(100.2)	// 10M band
69	#define ForwCal14MHz	(100.3)	// 14M band
70	#define ForwCal18MHz	(100.7)	// 18M band
71	#define ForwCal21MHz	(101.5)	// 21M band
72	#define ForwCal24MHz	(101.6)	// 24M band
73	#define ForwCal28MHz	(102.3)	// 28M band
74	#define ForwCal50MHz	(115.2)	// 50M band

1-3. 設定箇所-3 (LCD SWR の表示について)

下記の 77 行～86 行目のカッコ内の数値を調整する必要が有ります。25Ω と 100Ω のダミーロードが有れば最適の調整ができるのですが、筆者は持っていないので少し荒いやり方ですが、疑似的な調整で終わらせています。数値は送信機の表示に合わせて同じ値になるように補正しています。調整は各 BAND で行います。各 BAND で 10W 程度を入力してください。各バンド内で SWR が 2.0 程度の周波数を探してください。その周波数で補正します。(電波が出ますのでなるべく短時間で終わらせてください)

この時の LCD 出力表示が各 BAND とも送信機の SWR と同じになるように変更してください。

ほぼ同じ数値なら修正は不要です差が大きい時のみ行ってください。

77	#define ReflCal19MHz	(91.0)	// 1.8M band
78	#define ReflCal35MHz	(93.0)	// 3.5M band
79	#define ReflCal7MHz	(95.0)	// 7.0M band
80	#define ReflCal10MHz	(98.0)	// 10M band
81	#define ReflCal14MHz	(98.0)	// 14M band
82	#define ReflCal18MHz	(110.0)	// 18M band
83	#define ReflCal21MHz	(125.0)	// 21M band
84	#define ReflCal24MHz	(137.5)	// 24M band
85	#define ReflCal28MHz	(120.0)	// 28M band
86	#define ReflCal50MHz	(140.2)	// 50M band

1-1 の PTX 値が 1-2 の補正を行ったため若干ずれているはずです。再度 1-1、1-2 の補正を数回繰り返して行ってください。

1-4. 設定箇所-4 (LCD 周波数の表示について)

下記の 439 行の数値を調整する必要が有ります。MODE ボタンを 2 度押すと LCD の最下段に TX freq が表示されます。調整は 28MHz BAND で行います。10W 程度を入力してください。TX freq の右に送信周波数が表示されますのでこの周波数の差を補正します。前述しましたがカウンタ部は Nano 内蔵の 16MHz クロックを基準に計測しています。このクロックはそれほど正確では有りません。読み取りの誤差をプログラムで修正します。私の Nano での誤差率が【1.000897】でした。送信周波数と同じになるようにこの数値を補正してください。Nano のクロックは簡易な回路で作られています。熱に対しても変化しますので少しエージングしてから調整してください。この調整を怠ると BAND 切り替えがうまくいかない時が有ります。

```
439 float Freq_D = (count*16/100*1.000897);
```

1-5. 設定箇所-5 (LCD インターバル表示について)

下記の 138 行から 150 行のコールサイン等を変更して下さい。電源投入時 LCD の最初の表示はコールサインが 3 度断続して表示されます。御自身のコールサインと都市名に変更して下さい。不要でしたら 138 行から 154 行まで削除して下さい。

```
137 lcd.setCursor(0,1);  
138 lcd.print(" JA7FVT SENDAI ");  
139 delay(500);  
140 lcd.clear();  
141 delay(300);  
142 lcd.home();  
143 lcd.setCursor(0,1);  
144 lcd.print(" JA7FVT SENDAI ");  
145 delay(500);  
146 lcd.clear();  
147 delay(300);  
148 lcd.home();  
149 lcd.setCursor(0,1);  
150 lcd.print(" JA7FVT SENDAI ");  
151 delay(500);
```

以上です。

2024 年 10 月 28 日 (c) JA7FVT