FX.25 対応トランシーバ（試作）

May 3,2022　JK1MLY

基板のミス

S3(G0)と S4(EN)が近い

（想定しているのは 2pin タイプ;秋月 P-08078）

対応例：スイッチは使わない S4 側に 1uF とかを実装。S3 は未実装

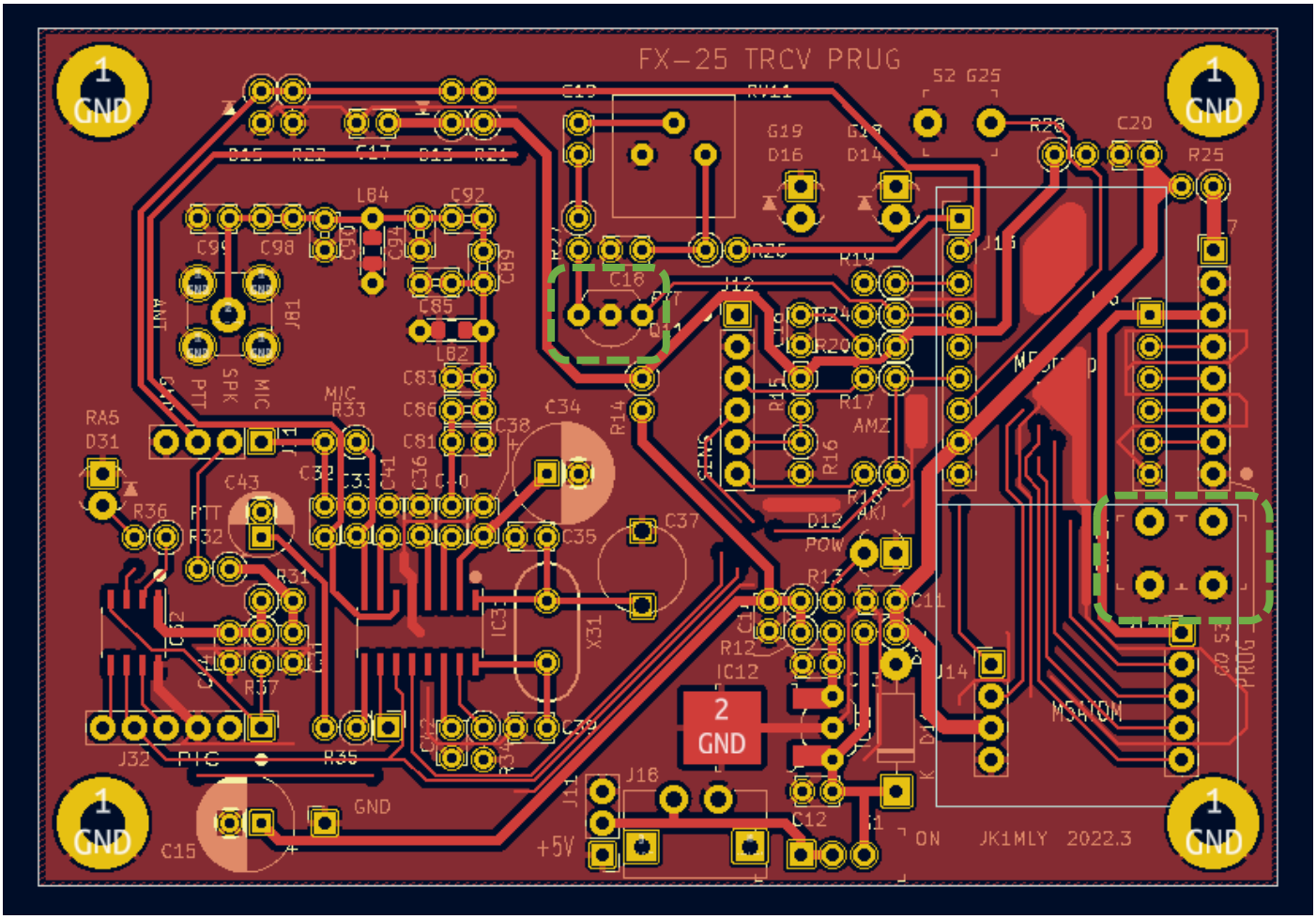
Q11 のピン配置が違う

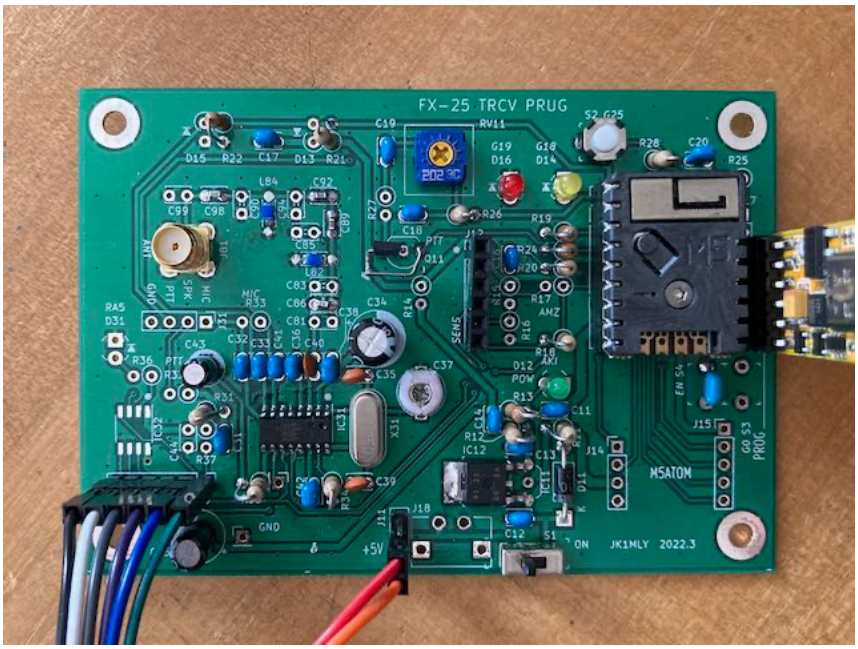
（想定しているのは DTC114EL;秋月 I-12467）

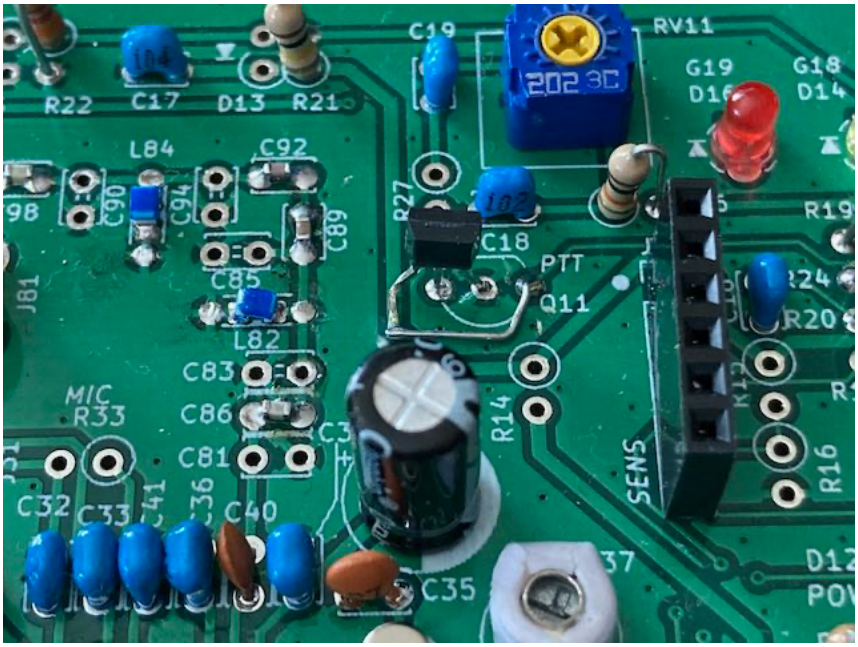
対応例：PTT は使わない　ずらして実装・表にジャンパ

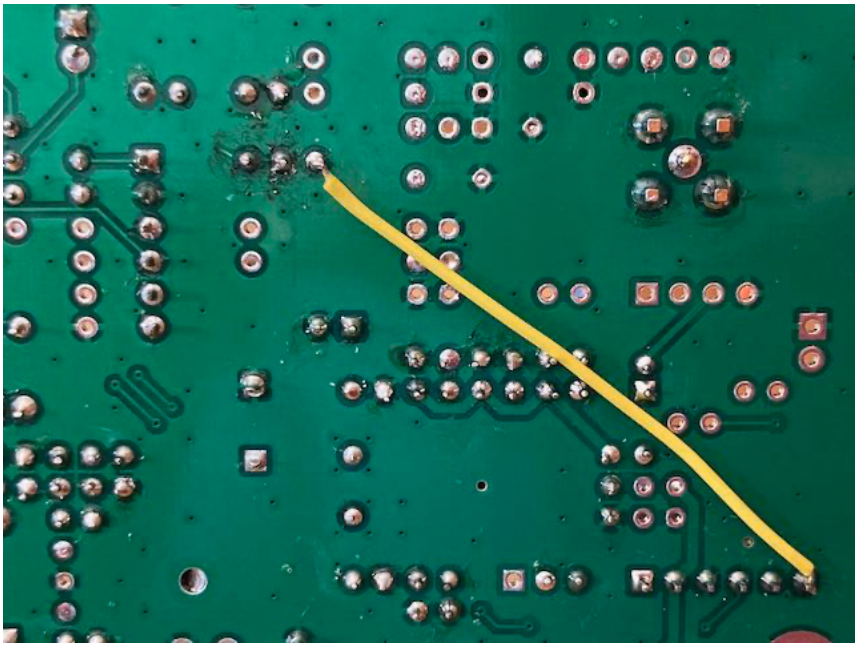
PIC の PTT への配線が無い

対応例：PIC 無しで動かす　Q11～J32 を裏でジャンパ




FX-25 TRCV PRUG
M5ATOM
JK1MLY 2022.3


RV11
C19
R22
C17
D13
R21
L84
C92
C18
PTT
Q11
R14
SENS
R33
MIC
C32
C33
C40
C35



TXD と RXD が逆

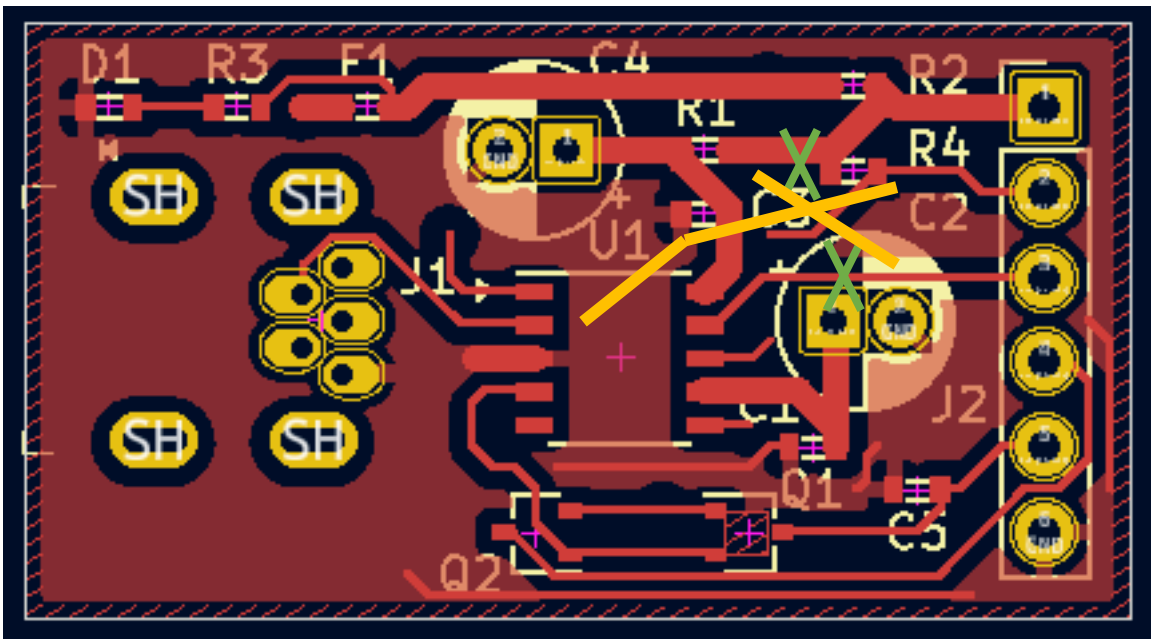
対応例：できたものを買う（秋月 M-17240　スイッチサイエンス /マルツなど M5STACK-A105）

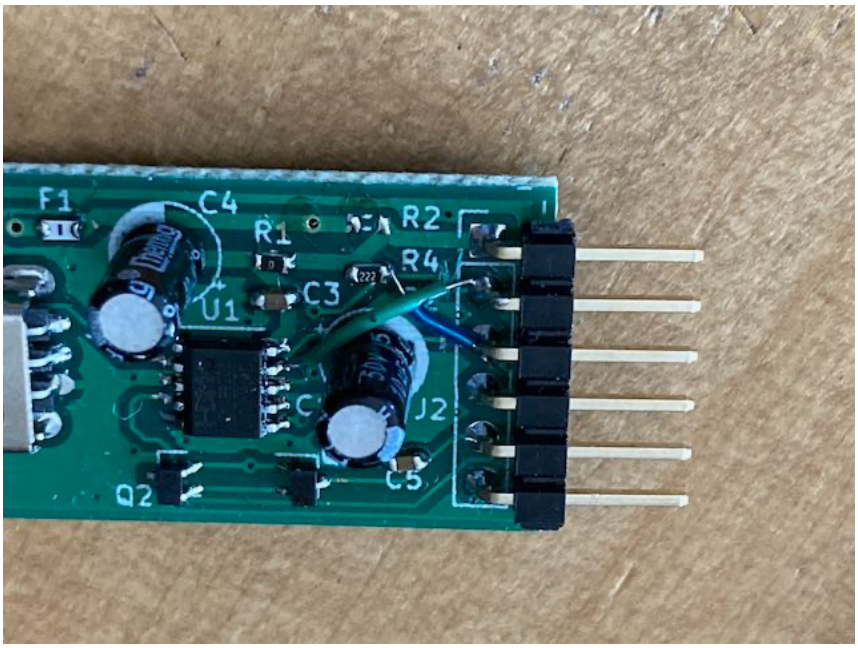
改造：コネクタの近くでパターンをカット R4 から 3pin　U1 9pin から 2pin にジャンパ

G0/EN の制御ができない

RTS/DTR の制御ができないっぽい（IC 出力が変化してない）

対応例：Q1,Q2 を実装せずシリアル変換だけ使用





基板のアセンブリ

以下に関する部分は実装の有無が変わる場合がある
- RF のフィルタ部分は評価を行っていない（定数が変わるかも）
- SOP の PIC を実装しての評価を行っていない（PicKit 接続ラインから相当回路）
- トランシーバ IC の制御が M5Stamp から出来ていない（ソフトを作ってない）
- センサモジュールの制御が M5Stamp から出来ていない（ソフトを作ってない）

部品リスト（作成中）

部品実装について

回路をチェックするため次の部品は後から実装
- R11 M5Stamp または Atom への電源供給
- R12 レギュレータからの電源供給
- R31 トランシーバ部への電源供給
- C40　トランシーバ IC の RF 出力

レギュレータは SMD タイプを推奨
手持ち部品を使えるよう TO-92 や TO-220 も実装できるパターンだが推奨は TO-252
ピン配置が違うレギュレータもあるので注意
裏面から半田が流れているのが確認できれば穴に半田を流さなくても良い

CPU モジュールは M5Stamp か Atom どちらかを実装
ただし Atom を使う場合は PIC と連携できないので、トランシーバ IC を ESP32 から制御が必須となる
M5Stamp の場合、スタック用のコネクタのモジュールへの付属品は 5mm タイプ
秋月は 3.5mm タイプの扱いのみ　使うならソケット(C-03138)とピンヘッダ(C-02900)の両方を変更
コネクタにモジュールを実装して半田付すると微妙な曲がりとかが出ないので良い

センサモジュールは秋月（K-09421 在庫無しが継続 ）と Amazon などに対応
ピン配置が違うため SCL を R17, R18 どちらに実装するかで切り替える
I2C-ADR を設定する場合は R15,R16 の実装で切り替える
それ以外の場合は Grove コネクタに接続する

低い部品から実装しないと後で大変
IC や RF フィルタの SMD など背の低い部品を実装
コンデンサやスイッチなどの 3mm 程度の部品を実装
抵抗など残りの部品を実装

基板の動作確認

S1 は OFF（ON のシルクと逆な方)、R11 と R12 と R31 は未実装、M5Stamp/Atom は未実装で確認

ショートのチェック

J11 の 1pin と 2pin
IC11 の 1pin と 3pin
GND と R11　R11 の両方の端子を確認
GND と R12　R12 の両方の端子を確認
GND と R31　R31 の両方の端子を確認

電圧の確認

J11 または J18 から 5V を供給

R11 のシルクで丸が無い側の電圧を確認　S1 を ON すると 5V±0.2V 程度なこと
R12 のシルクで丸が無い側の電圧を確認　S1 を ON すると 3.3V±0.1V 程度なこと

R11 を実装

M5Stamp または Atom の 5V の電圧を確認　S1 を ON すると 5V±0.2V 程度なこと

R12 を実装

S1 を ON すると D12(LED-G)が点灯
R14 のシルクで丸が無い側の電圧を確認　S1 を ON すると 3.3V±0.1V 程度なこと
M5Stamp の 3.3V のピン(3V3)に電圧が出てないことを確認
R31 のシルクで丸が無い側に電圧が出てないことを確認

PIC への PicKit3 接続（IC32 を使う場合のみ）

（予めコンパイルできる環境を構築してある前提）
J32(PIC)に PicKit3 などを接続
MPLAB IDE で Power 供給は基板側（チェック無）なことを確認
S1 を ON すると D12(LED-G)が点灯
Program Device でプログラムできることを確認

CPU モジュールの実装

（予めコンパイルできる環境を構築してある前提）
S1 を ON すると D12(LED-G)が点灯
S1 を OFF
J17(PROG)にダウンロードケーブルを接続
新しく認識されたシリアルポートを控える
Idf.py コマンドでプログラムできることを確認

```
idf.py -p ポート -b 115200 flash
```

トランシーバ IC

フィルタの特性を確認

C81 の場所にネットワークを接続するためのケーブルを付ける
J81 との間で 430MHz±10M の S21 を測る
430MHz から 432MHz ぐらいのロスが小さいことを確認する
帯域が広すぎ or 狭すぎ、または 1 次の傾斜があったら C88 と C91 で微調整
周波数が高すぎ or 低すぎ、C82 と C95 で微調整
仮に付けたケーブルを外す
C40 を実装する

R31 を実装して電圧を確認

S1 を ON すると D12(LED-G)が点灯
R14 のシルクで丸が無い側の電圧を確認　S1 を ON すると 3.3V±0.1V 程度なこと

トリマで周波数を調整

C37 で XO(16pin)が 21.25MHz になるよう C37 を調整
連続で送信できるなら 430M または 144M の電波の周波数で調整しても良い
高すぎて合わせられない場合は C35 を 22pF にする
まだ高い時には C39 を 22pF にする
さらに高いなら 5pF 程度ずつ順に減らしていく
低すぎて合わせられない場合は C39 を 33pF にする
まだ低い時には C35 を 22pF にする
さらに低いなら 5pF 程度ずつ順に増やしていく

送信出力の確認

実測できなくても計算値で保証認定は通る
連続で送信できるなら J81 にパワー計を接続して測る
2mW 程度の出力が得られていることを確認

送信スペクトラムの確認

実測できなくても計算値で保証認定は通る
連続で送信できるなら J81 にスペクトラムアナライザを接続
第 5 高調波あたり（携帯を考慮して 430M なら 2.2G、144M なら 900M）まで測れると良い
20dB ぐらい取れていれば問題ないハズだが気になる場合は細かく見る

ソフト

ESP32 の開発環境

MacOS 12.3.1 で自分の環境の場合のメモ
git で元になるファイルを取ってきてオリジナルが書けるまでに行ったこと
VSCode をインストール
拡張機能で「Espressif IDF」をインストール
「ESP-IDF Examples」で「blink」をコンパイルしてみる
GUI で開発しても良いが、これで関係するものは自動で入る＋設定できると思う
参考にしたサイトは下記

https://zenn.dev/thorie/articles/4b65e6d8f29a3f

コンパイルして書き込めるかの確認は以下(GUI は使ってない)
オリジナルの readme に書かれていることに沿って作業
https://github.com/amedes/ESP32TNC

```
ls /dev/cu*
・・・ポートを確認
git clone https://github.com/amedes/ESP32TNC.git
cd ESP32TNC
. ../export.sh
idf.py menuconfig
・・・とりあえず Atim あたりで設定
idf.py -p ポート -b 115200 flash
idf.py -p ポート -b 115200 monitor
・・・起動時のメッセージが出てきたら大丈夫
```

PIC の開発環境

MacOS 版だとダウンロードするだけで使えた

https://www.microchip.com/en-us/tools-resources/develop/mplab-x-ide

参照先

Github 上に全て置かれているので、以下から入手することができる

ESP32(master)

https://github.com/amedes/ESP32TNC

ここが本家となる。FX.25 としての機能開発はココに最新が入る

ESP32

https://github.com/jk1mly/ESP32TNC/tree/qrp-tnc

無線機を制御の追加やピンの変更を行うためフォーク

今までに定義されている基板とピンが違うので M5StampQRP を増やしてある

PIC

https://github.com/jk1mly/fm-trcv/tree/qrp-tnc

QRP 無線機の制御を PIC に残す場合のブランチ

ピンの割り振り、デバイスを変える必要があるため分けてある

トランシーバ基板

https://github.com/jk1mly/pcb-qrp-tnc

BK4802 と M5Stack を使った基板

ダウンローダ基板

https://github.com/jk1mly/pcb-ch340k/tree/dl

CH340K を使った基板を幾つか考えていてダウンローダ用に分けたもの