FpcROUTE

A File Translator for <u>FreePCB</u> and <u>Freerouter</u>

User Guide

Version 1.300

FreePCB   FpcROUTE   Freerouter

Motor.fpc

Motor.dsn

Motor_routed.fpc

Motor.ses
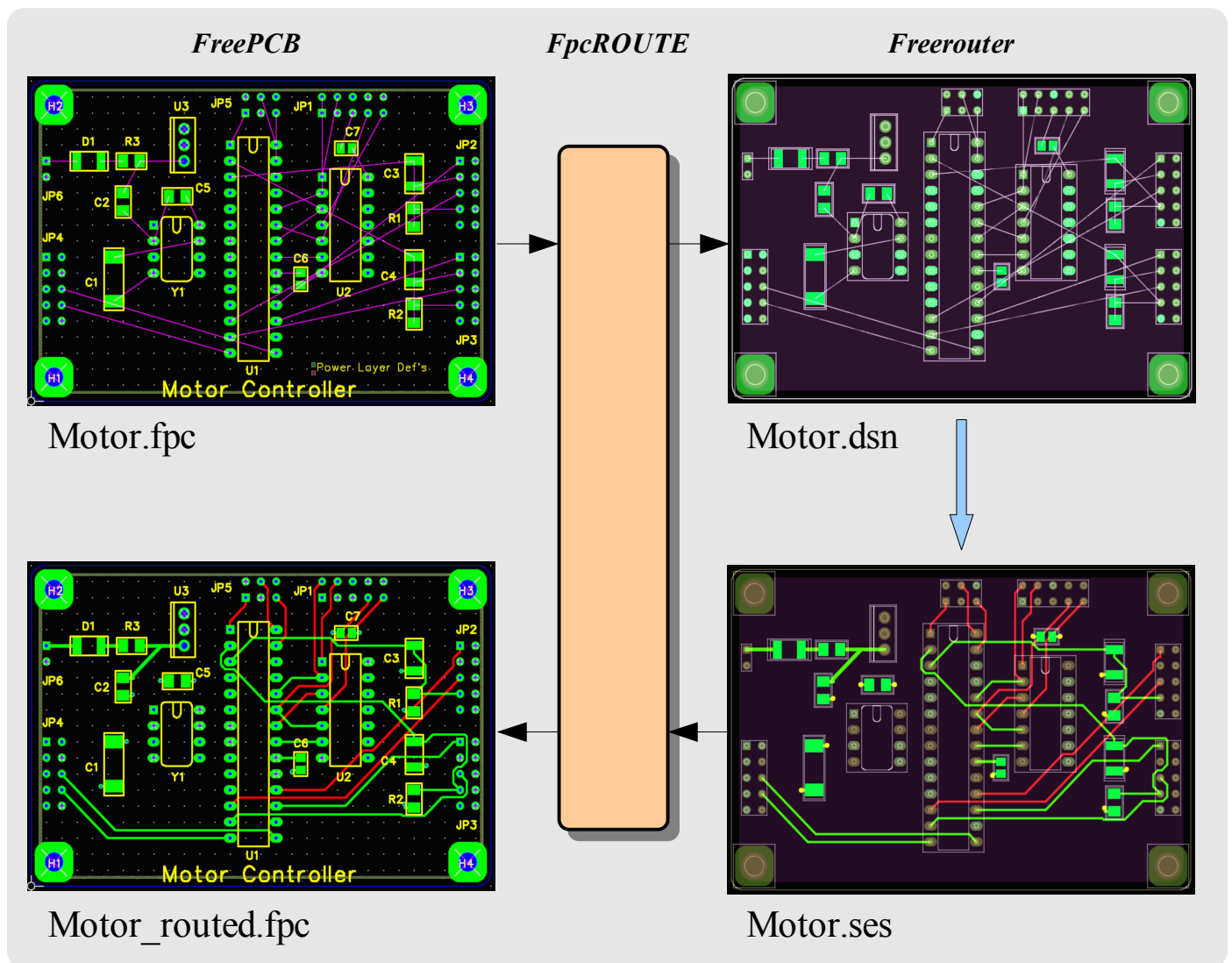
Note: this document has been reformatted for two-sided print.

Contents

FpcROUTE End User License Agreement (EULA)

General

FpcROUTE Software is distributed as Freeware. You may use the Software on any number of computers for as long as you like. The Software is NOT Public Domain software. You are allowed to freely distribute the Software, but Bruce Parham retains ownership and copyright of the Software in its entirety. You may use and/or distribute the Software only subject to the following conditions:

1. You may not modify the program or documentation files in any way.

2. You must include all the files that were in the original distribution.

3. You may not reverse engineer, decompile, or otherwise reduce the Software to a human perceivable form.

4. You may not sell the Software or charge a distribution fee, except to recover the media costs.

5. You understand and agree with this license and with the Disclaimer of Warranty and the Limitation of Liability printed below.

6. The Software may be bundled and distributed together with other software products provided the same conditions apply.

Agreement to this license

By using, copying, transmitting, distributing or installing the Software, you agree to all terms of this license. If you do not agree to all of the terms of this License, then do not use, copy, transmit, distribute or install the Software and immediately remove the Software from your storage device(s).

Warranty disclaimers and liability limitations

The Software and related documentation are provided "as is", without warranty of any kind. None whatsoever. Bruce Parham disclaims all warranties, express or implied, including, but not limited to, warranties of design, merchantability, or fitness for a particular purpose or any warranty of title or non-infringement. Bruce Parham does not warrant that the functions contained in the Software or documentation will meet your requirements, or that the operation of the Software will be error-free, complete, or that defects in the Software or documentation will be corrected.

Limitation of Liability

Under no circumstances shall Bruce Parham be liable for any lost revenue or profits or any incidental, indirect, special, or consequential damages that result from the use or inability to use the Software or related products or documentation, no matter what legal theory it is based on.

Copyright and trademarks

FpcROUTE is Copyright © 2007-08 by Bruce Parham. All Rights Reserved.

Introduction

FpcROUTE is a Win2K/WinXP console file translator application with integrated support in *FreePCB*. As such, it can be launched form within *FreePCB* or directly from a command shell.
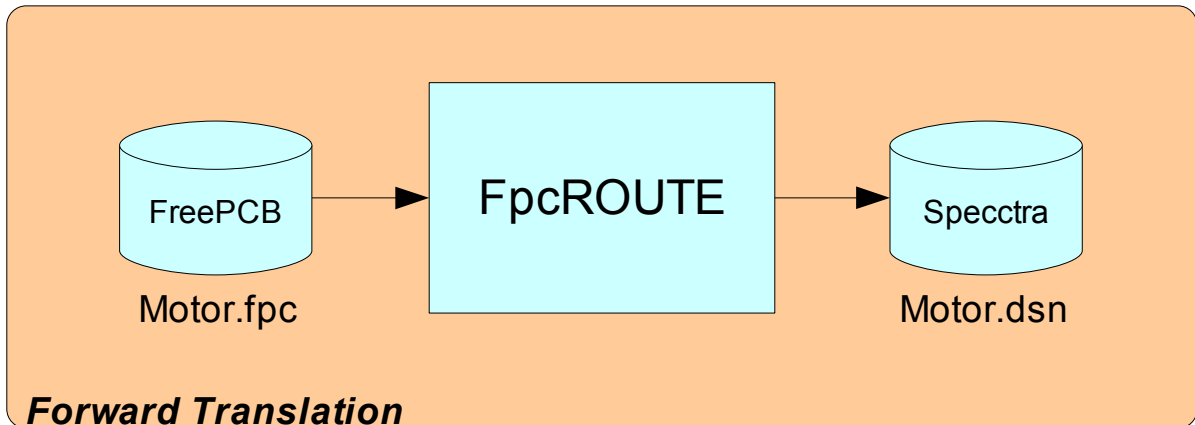
FpcROUTE will translate a *FreePCB* board into an *Specctra* design file suitable for import into Alfons Wirtz's web based *Freerouter*. Prerouted wiring, FromTos, keepouts and power planes can be defined within *FreePCB* to control how the board is autorouted.
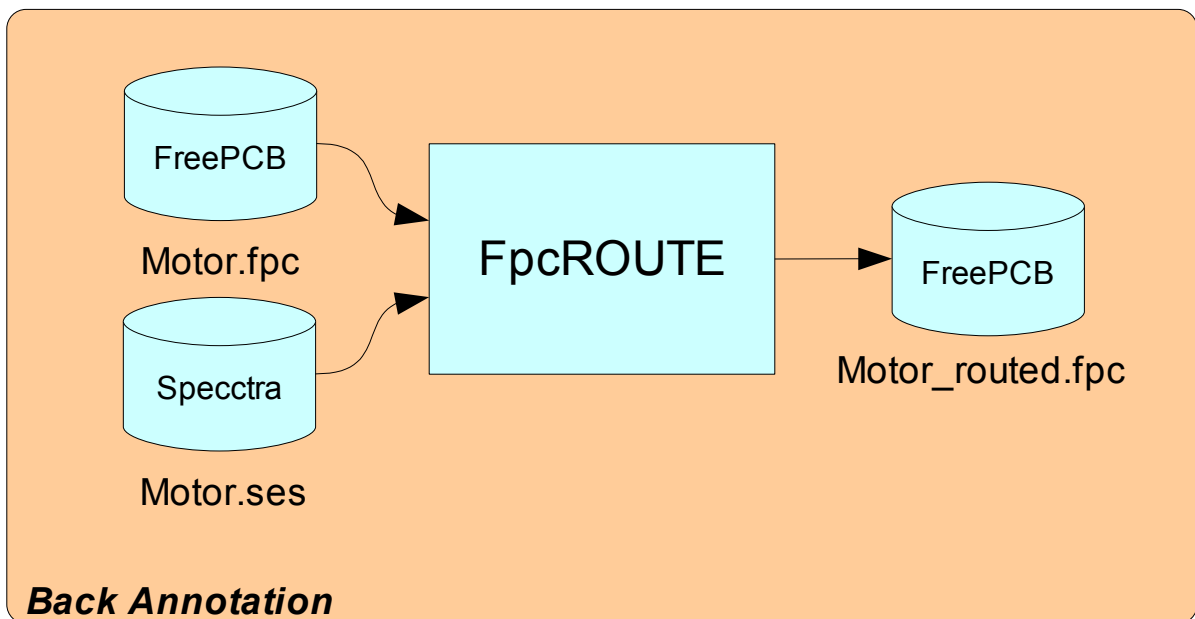
FpcROUTE will also perform the reverse translation by converting a *Specctra* session file, output by *Freerouter*, into a routed *FreePCB* board file.

Installation and Setup

FpcROUTE is distributed as a set of files in a zip archive that includes the executable, the user guide and a selection of example files. The *FpcROUTE* executable is also included in the latest *FreePCB* Combined Updates.

To install *FpcROUTE* for stand alone use, extract the file **FpcROUTE.exe** to any handy directory such as C:\Program Files\ FpcROUTE\ or C:\Program Files\FreePCB\bin\. If **FpcROUTE** is to be used from within **FreePCB** to export/import files, then **FpcROUTE.exe** must be placed in the same directory as the **FreePCB** executable. This is usually C:\Program Files\FreePCB\bin\.

For stand alone use, if the path to **FpcROUTE.exe** is not already included in the environment **PATH** variable, it should be added. The **PATH** variable does not need to be set if **FpcROUTE** is only going to be used from within **FreePCB.**

For those unfamiliar with setting environment variables, assume for example that **FpcROUTE.exe** is installed in the C:\Program Files\FreePCB\bin directory:

- Open the **System Properties** form by

    - Right clicking on **My Computer** and clicking on **Properties**.
  or
    - Open the **Control Panel**.
    - Open the **System** tool.

- Click the **Advanced** tab and open the **Environment Variables**.

- Scroll the variable list down to and highlight the **path**= entry and click **EDIT**.

- Click in the value field to clear the highlight.

- Scroll to the end of the field and add **;C:\Program Files\FreePCB\bin** to the existing value.

- Click **OK** to exit the edit window.

- Click **OK** to save and exit the Environment Variables window.

- Click **OK** to exit System Properties.

Operation

Integrated with FreePCB
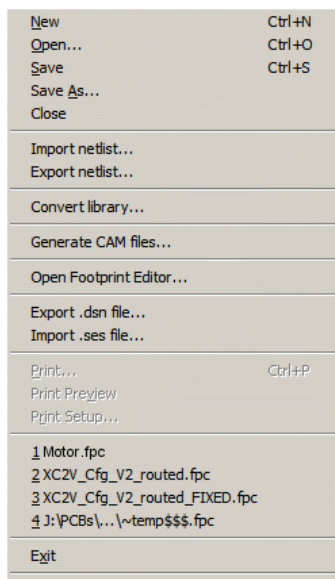
Fig 1. File Menu

The simplest way to use **FpcROUTE** is from within **FreePCB** by using the *Export .dsn* and *Import .ses* file commands found in the **File** pull-down menu.
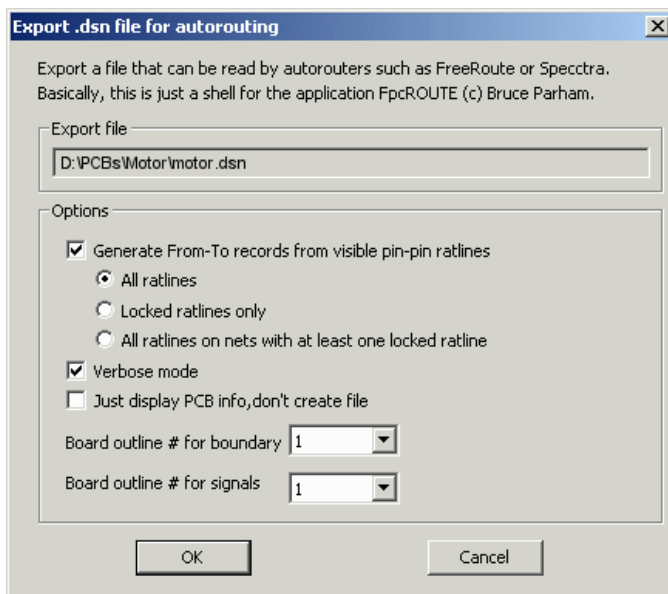

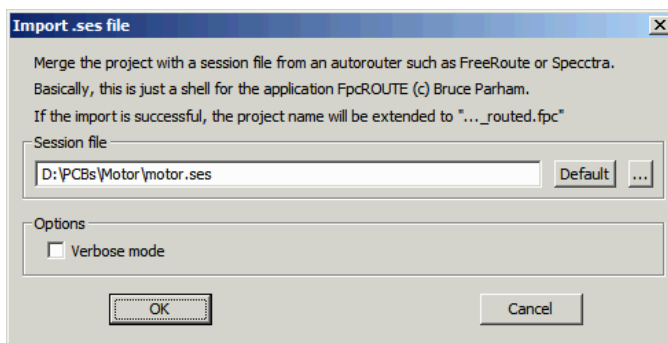*Fig 2. **Export .dsn file** window*

The **Specctra Design File Export** and **Import** windows provides a simple means for controlling **FpcROUTE**.

*Fig 3. **Import .ses file** window*

Command line Operation

FpcROUTE can also be operated form a command line. To do this, open a command shell[1], change the current directory[2] to where the *FreePCB* board file is stored and, assuming the board file is names "Motor.fpc", at the command prompt, type: `fpcroute motor` to begin the conversion. *FpcROUTE* will load the "Motor.fpc" board file and build a board database. Once the database is complete, a *Specctra* design file named "Motor.dsn" is created. The new design file is ready to be loaded into *Freerouter*.

When the board is routed and adjusted to your liking in *Freerouter*, export a *Specctra* session file named, in this case, "Motor.ses". Back in the command shell, type: `fpcroute -b motor` to begin the back annotation process[3]. Again *FpcROUTE* will load "Motor.fpc" and build a board database. Then the session file "Motor.ses" is loaded and the database is updated with any new component placement and/or wiring information. Finally, *FpcROUTE* creates a new *FreePCB* board file from the updated database named "Motor_routed.fpc".

Command Line Syntax

The command line syntax was designed for simple yet flexible operation. The full syntax consists of the program name followed by optional switches and one or two file names:

   `C:> fpcroute [switches] [path1]name1[.ext1] [[path2]name2[.ext2]]`

If only one file name is entered, it will be used for both the *FreePCB* and *Specctra* files. If the file name includes an extension, the name will be used <u>as entered</u> for the *FreePCB* source file; the *Specctra* name will use the base name with the extension changed to ".dsn" or ".ses" as appropriate. If the file name does not include a extension, ".fpc", ".dsn" or ".ses" will be added as needed.

If two file names are present, they will be used as entered, with default extensions added as needed. The first name refers to the *FreePCB* file and the second used for the *Specctra* file.

File names may optionally include any valid, existing absolute or relative path. If the path includes spaces, the whole name must be enclosed in quotes ("). If the path does not exist, it will not be created and *FpcROUTE* will exit with an "Unable to open..." error. If a path is not included with the name, the current directory is used.

The *FreePCB* file created during back annotation uses the source *FreePCB* name and extension with "_routed" appended to the base file name.

When started, *FpcROUTE* <u>always</u> loads the named *FreePCB* file and builds an internal pcb database. What happens next depends on what, if any, control switches are present. A small number of optional command line switches are available. Each switch is specified by a single character preceded by a dash (-) and may be followed immediately, <u>without any spaces</u>, by one or more text or numeric values. If embedded spaces are needed in the value, the whole field including the dash and switch character must be enclosed in quotes. The switch characters are <u>not</u> case sensitive.

Some equivalent parameter entries with and without embedded spaces:

    `C:> fpcroute -u1,2 motor`

    `C:> fpcroute "-U1 2" motor`

---

1   Windows Task Bar: Start→Run→cmd→OK.
2   cd command.
3   See pg 32, Known Bugs and Issues

Currently available switches:

-B   ***Back*** annotate route and placement data from sesfile to fpc_routed file.

-Fn   Include fpcfile *rat-lines* as design file net ***FromTo*** records

-H | -?   Display a short switch and usage ***Help*** summary and quit

-I   Display fpcfile statistics ***Info*** and quit

-Un1[,n2]   Boundary ***oUtline*** polygon select: n1 = pcb & signal [n2 = signal]

-V   Enable ***Verbose*** debug output

–B : Back Annotate

Back annotate is the main mode control switch for *FpcROUTE*. When used, it causes *FpcROUTE* to load a *Specctra* session file, update the *FreePCB* board database and create a new *FreePCB* file from the database with updated wiring and placement data. When not used at startup, F*pcROUTE* performs forward conversion by creating a *Specctra* design file from the loaded pcb data.

If the verbose switch is also used, a multi-pass wiring analysis listing is sent to `stdout`. This listing can be redirected to a file by using the standard command line redirection "`>`" syntax.

–Fn : FromTo Control

Control the generation of net **FromTo** records in the *Specctra* design file. By default *FreePCB* rat-line data is not included in a design file because the router automatically recreates the same net topology when the design is loaded. In the case where the rat-line topology has been edited in *FreePCB* to create a non-default pin-to-pin pattern, **FromTo** records can be used to convey the modified layout information to the router.

Four modes of operation are available:

       -F0 = **FromTo** routing off (default state)
       -F1 = add **FromTo** records <u>only</u> for <u>locked</u> rat-line segments
       -F2 = add **FromTo** records <u>only</u> for <u>visible nets with locked rat-line segments</u>.
       -F3 = add **FromTo** records for <u>all visible rat-lines on all nets</u>.

In the first active (-F1) case, only <u>locked, pin-to-pin rat-lines</u> are added as **FromTo** records. *The presence of a **FromTo** record within a net causes the router to ignore the normal netlist data for that net and only route the **FromTo** defined paths. A*ny unlocked rats will be ignored.

In the second active (-F2) case, any net, with a <u>locked</u> pin-to-pin rat-line segment, will have <u>all visible segments</u> added to the design file as **FromTo** records.

In the last (-F3) case, <u>all visible pin-to-pin rat-lines</u> of <u>all</u> nets are added as **FromTo** records. Power fanouts should be added by hand prior to autorouting.

In both of the last two cases net <u>visibility</u> is an issue because *FreePCB* does not export rat-line data for unlocked invisible nets. Also, only ***pin-to-pin*** rat-lines can be converted to **FromTo** records; ***trace-to-trace*** and ***trace-to-pin*** rat-lines are ignored.
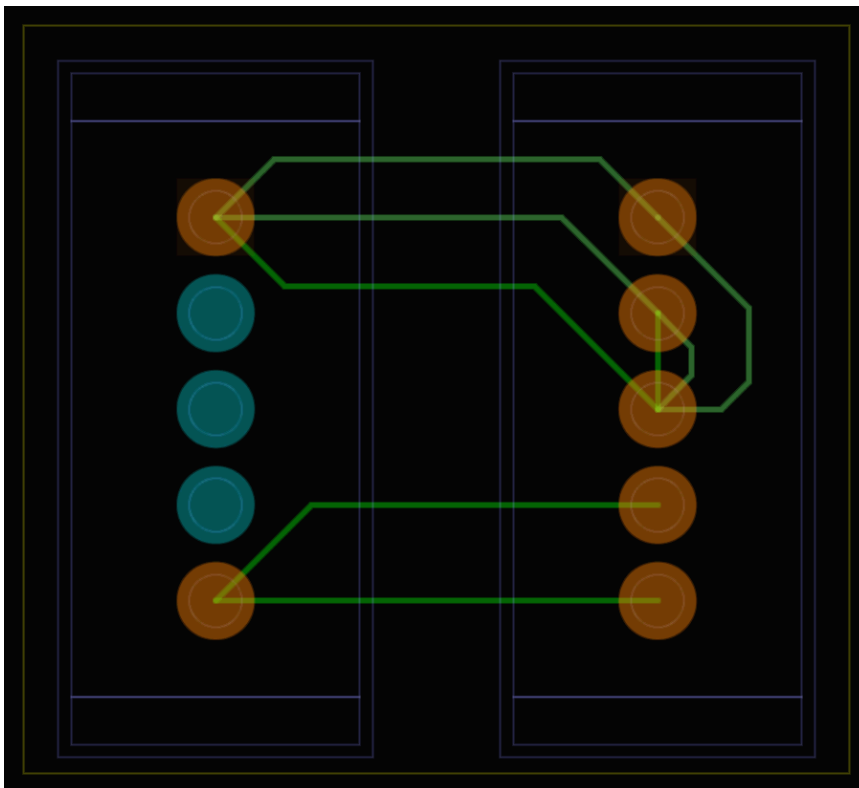
<u>Some care should be exercised with this option</u> because a **FromTo** is interpreted by the router to mean ***must connect pin-to-pin***. Segments that would normally be converted to power plane fanouts will instead be routed pin to pin. In addition, **FromTo** specified traces will not use copper sharing tees.

Fig 4. Hand Placed Rat Lines
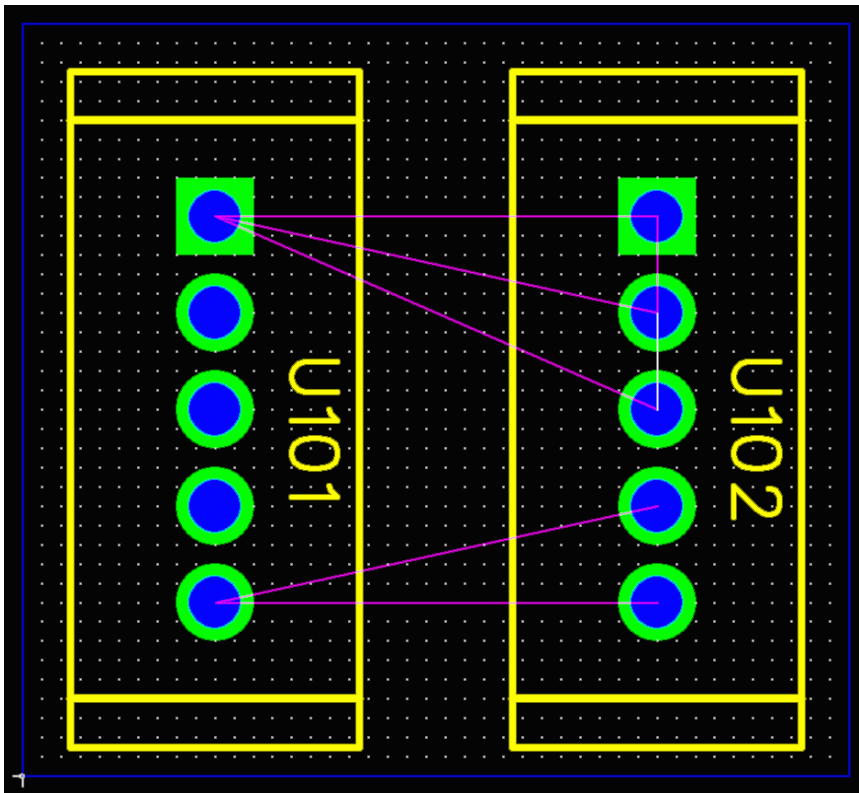
The default rat-lines were deleted and replaced manually. Note that U102 pin 3 connects to pin 1 and pin 2 *twice*. The net section of the dsn shows how each rat-line becomes a FromTo:

```
(net "Net 1"
    (pins
        "U101"-"1"
        "U102"-"1"
        "U102"-"2"
        "U102"-"3"
    )
    (fromto "U101"-"1" "U102"-"1")
    (fromto "U101"-"1" "U102"-"2")
    (fromto "U101"-"1" "U102"-"3")
    (fromto "U102"-"1" "U102"-"3")
    (fromto "U102"-"2" "U102"-"3")
    (fromto "U102"-"2" "U102"-"3")
)
(net "Net 2"
    (pins
        "U101"-"5"
        "U102"-"5"
        "U102"-"4"
    )
    (fromto "U101"-"5" "U102"-"4")
    (fromto "U101"-"5" "U102"-"5")
)
```


Fig 5. Routed and Cleaned

Once routed, the manually created topology is evident. Note how the router has preserved the redundant connections between pins 2 and 3 of U102.

–H or –? : Help

Display a short help screen and terminate.

–I : Information

Load the *FreePCB* board database and display summary information, list <u>Net Classes</u> and <u>Padstacks</u>. If the Verbose switch is also used, a recreated *FreePCB* file listing is sent to `stdout`. A design file is not created.

–Un1[,n2] : oUtline Select

The *Specctra* design file requires that the board limits be specified by a rectangular PCB outline and can also use a polygon outline to serve as a signal routing keepin. By default, the #1 board outline in the *FreePCB* board file is used for both design file outline figures. For boards with more than one outline, the outline select switch can be used to specify which outline or outlines are used. If only one parameter, *n1*, is present, the board outline specified by *n1* will be used for both the PCB and signal outlines. If both *n1* and *n2* parameters are present, *n1* will select the PCB outline and *n2* the signal outline. Note that the PCB outline rectangle is the bounding box of the selected board outline and is not normally visible in *Freerouter*. Also, the signal outline must be equal to or fully contained within the PCB outline. The signal outline is the outline visible in the router.

–V : Verbose Debug Output Enable

The verbose output switch can be used with the **–I** and **–B** switches to create debug listings directed to he `stdout` terminal port. These listings are mainly for software debug but may prove interesting or useful to the users.

Autorouting[4]

The autorouter can be started within FreePCB by clicking **Tools→FreeRoute autorouter...** or directly from the Freerouting web page at http://www.freerouting.net/.

Once the router is started, click **Open Your Own Design** and navigate to and open the DSN file. In a few seconds, the router **Layout** and **Select Parameter** windows will open. In the Select Parameter window, make sure both `board outline` and `fixed` are checked. (If the Select Parameter window does not automatically open, open it by clicking the tool bar Parameter→Select.)

To insure that the autorouter maintains proper clearance to the board outline, the outline clearance class need to be changed. Select the board outline by dragging a small box over a segment. With the outline selected, click **Clearance** in the menu bar and change the selected class from **Null** to **Outlines**.

Make any other parameter changes as needed and, once satisfied that everything is set, start the autorouter by clicking **Autorouter** on the tool bar.
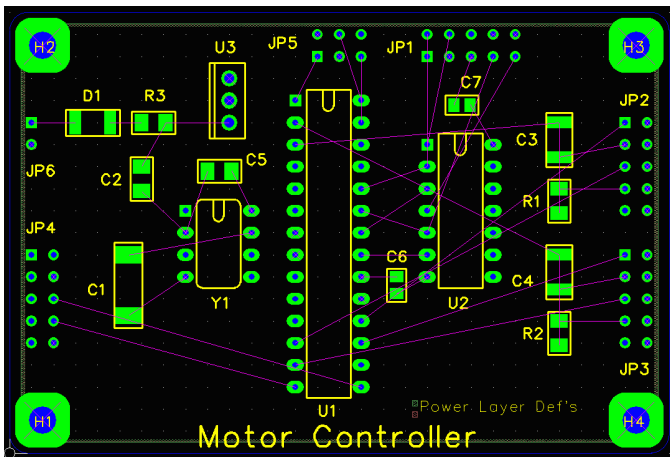

*Fig 6. **FreePCB** board*   Motor.fpc
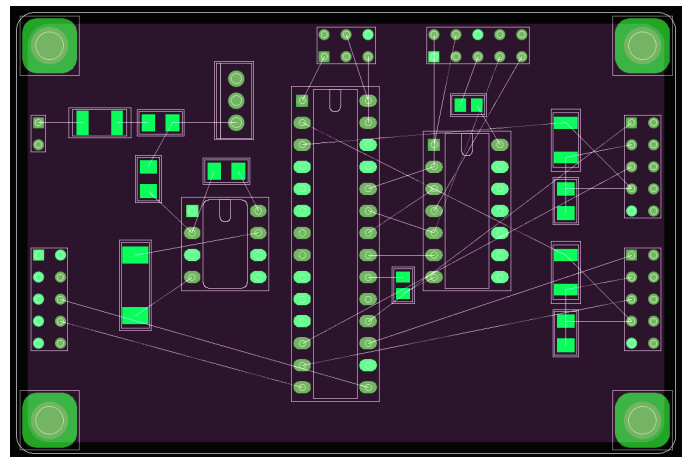
*Fig 7. **Freerouter** loaded*   Motor.dsn

Back Annotation

When the board is routed and edited to your liking in ***Freerouter***, export a ***Specctra*** session file named the same as the .dsn file but with the extension changed to .ses. Import the session file into ***FreePCB*** by clicking on ***Import .ses file...*** and selecting the newly created file. This will run ***FpcROUTE*** in the back annotate mode and load the routed board into FreePCB.

*Fig 8. **Freerouter** done*   Motor.ses

*Fig 9. Routed board in **FreePCB***   Motor_routed.fpc

---

4   A functioning internet link is required to start the ***FreeRoute*** autorouter.

PCB Requirements for Autorouting and Back Annotation

For successful file translation, a few simple rules need to followed:
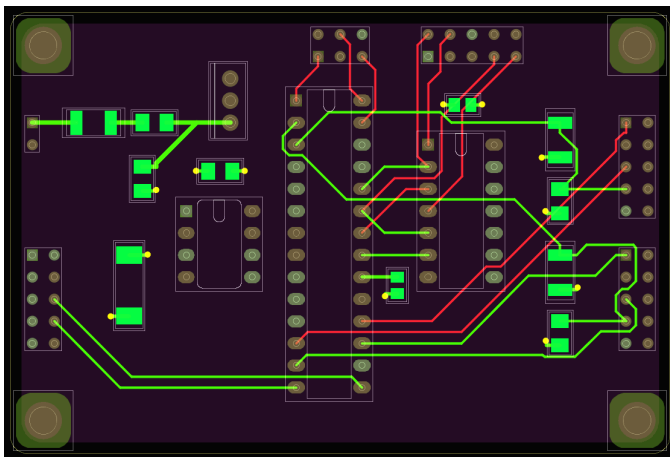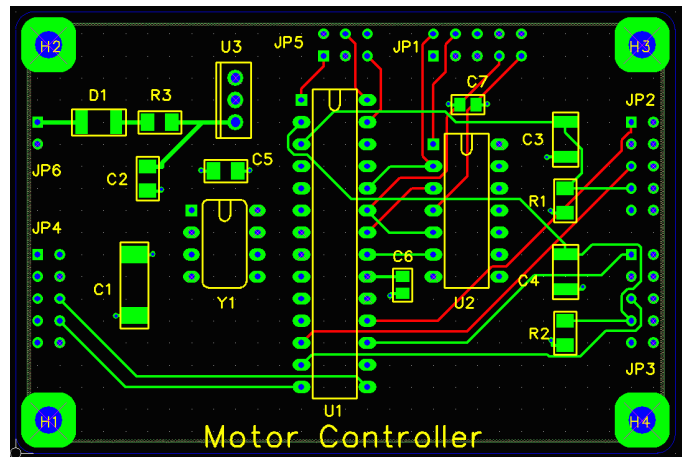
1. The board must have at least one outline and the outline must contain all active pins to be routed. If a part with active pins (pins with nets assigned) falls outside the outline, the pins outside the outline will be unroutable.

2. If more than one outline is used, the signal outline may touch but not fall outside the PCB outline bounding box. If the signal outline does exceed the PCB outline bounding box, translation is halted with an error message.

3. In *Freerouter* parts can be moved and rotated in 1° increments. *FpcROUTE* will update position and angle changes during back annotation but *FreePCB* only supports 90° rotation increments so, if a part is rotated any angle other than a multiple of 90°, the back annotation process will fail with an error message.

4. Do not define any new or edit any existing pins or vias in the router. The current version of *FpcROUTE* does not support back annotation of *padstacks* from the session file. New padstacks are ignored and an "Unknown Padstack" error will be generated when any new padstack is encountered within the wiring. *This behavior may change in a future version*.

PCB Recommendations for Autorouting

1. Parts that are <u>Glued</u> (position locked) in ***FreePCB***, will become <u>System Fixed</u> in ***Freerouter*** with no way available to unfix them. The Fix/Unfix operation in ***Freerouter*** only works on <u>User Fixed</u> objects. If you think you may want to move a part in the router, make sure it is <u>Unglued</u>.

2. Define inner layers with power planes (copper areas) as <u>Power Layers</u> to have the router create fanouts and prevent the router from placing traces <u>in</u> the power planes[5].

3. Preroute any traces you want to have specific topologies in ***FreePCB*** before using the router. This especially applies to rat-lines used with **FromTo** routed nets[6]. All traces, vias and copper areas are preserved by ***FpcROUTE*** and will be included in the Specctra design file. Traces and vias become <u>User Fixed</u> objects which can be unfixed and modified. All copper areas are loaded as <u>System Fixed</u> and may not be changed.

    FreePCB has the ability to create pins and polygons with curved segments, ***Freerouter***, or any ***Specctra*** type router for that matter, does not. ***FpcROUTE*** approximates curved segments in pins, copper areas, area cutouts, outlines and footprint graphics by replacing the curved segment with a series of short straight segments. For this reason, copper areas no longer match their original area shapes and are not updated in the back annotation process, the original shapes are used instead.

4. Use Keepouts and Via_Keepouts to control router trace and via placement[7]. Areas defined as <u>Keepouts</u> can use cutouts like any other area.

5. If your design includes Text blocks on signal layers, cover the text with a keepout to prevent the router from running traces through the text and possibly creating shorts.

6. Invisible net rat-lines that are unlocked are not exported by ***FreePCB***. If router ***FromTo*** records are enabled, make sure the effected net rat-lines are visible or locked.

7. Run ***FpcROUTE*** on a newly created project to help identify netlist and footprint problems.

    Common errors:

    `Part <refdes> has unknown pin: <pin_name>` indicates an imported netlist referenced an invalid footprint pin name. The footprint for part <refdes> does not have a pin named <pin_name>. Names and reference designators <u>are case sensitive</u> and must match exactly.

    `Unknown RefDes: <refdes>` indicates a missing part. Again, check for case match.

---

5   See the next page for how to define **Power Layers**
6   See **-Fn** option comments on Pg 11
7   See the next page for how to define **Keepouts**

Reserved Net Names

The following net names, <u>all in upper case</u>, are reserved by FpcROUTE for special area and layer definitions and, <u>if used</u>, should be added to your Project→Nets table after loading your netlist:

1. **KEEPOUT**

2. **KEEPOUT_VIA**

3. **KEEPOUT_ALL**

4. **POWER_LAYER**

Keepouts

Freerouter can utilize two types of routing keepouts to control autorouter trace and/or via placement. To define a keepout in *FreePCB*, create a copper area on the desired layer and connected to one of the reserved keepout net names. Any DRC errors in *FreePCB* caused by these areas can be ignored; all keepout areas and nets are removed during back annotation. <u>Do not route any traces on these nets.</u>

Areas attached to net "KEEPOUT" will block both traces and vias on that layer only.

Areas attached to net "KEEPOUT_ALL" will block both traces and vias on all signal layers.

Areas attached to net "KEEPOUT_VIA" will block vias but not traces on that layer. Note that since *FreePCB* uses only through hole vias, a via keepout on any layer effectively blocks vias on all layers.

In addition to user defined polygon keepouts, circular keepouts are added to all purely mechanical pins, i.e. through hole pins without copper on any layer. These keepouts are attached to the footprints and will move and change sides with the footprint. The circular keepout radius is sized to the drill radius + the DRC "Hole to Pad or Trace" value.

Power Planes

Freerouter can use two types of copper layers: signal and power. Signal layers can be used for trace routing and may have user defined copper areas. Power layers, on the other hand, are not normally used for trace routing but may have any number of copper areas. By default, all layers are defined as type signal.

To define a layer as a power layer, place a copper area anywhere on that layer and attach it to the "POWER_LAYER" net. The location and size of the area is unimportant; it need not be within the board outline. Like keepouts, power layer defining areas are removed during back annotation.

Graphics

No differences other than line widths exist between the shapes displayed in **FreePCB** and **Freerouter**.

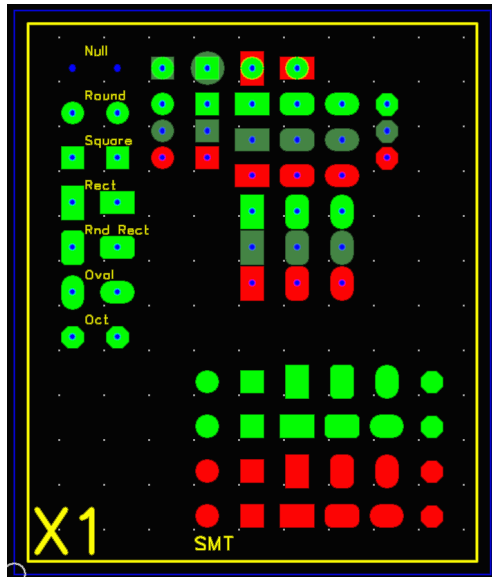FreePCB supports six different pad shapes for pins plus padless drilled holes. When **FpcROUTE** creates the
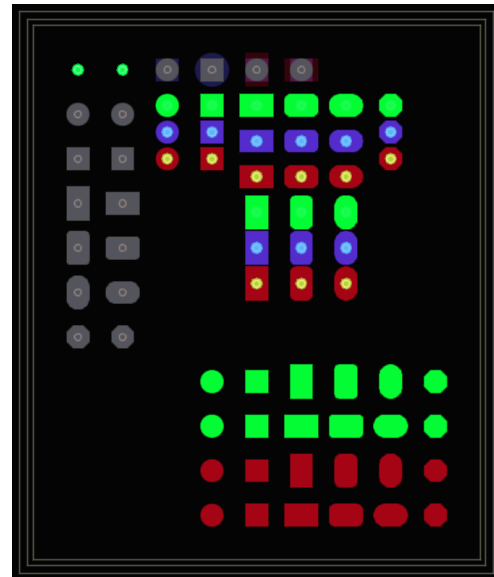

*Fig 10. Pads in **FreePCB***

*Fig 11. Pads in **Freerouter***

design files padstacks, round, square and rectangular pads are converted directly while other shapes are replaced with polygons.

In addition, for through hole pins, any pin that does not have copper defined, i.e., no pads, will have a circular keepout added with the keepout radius set to the drill radius plus the current *Hole to Pad or Trace* DRC setting. These keepouts become part of the component and will move with the part if it is moved or flipped to the other side.
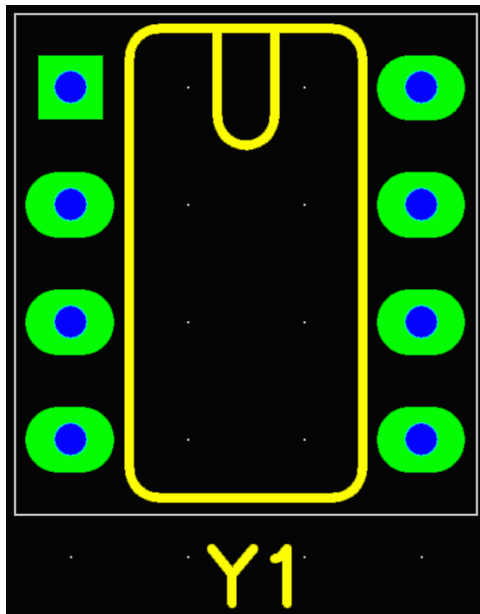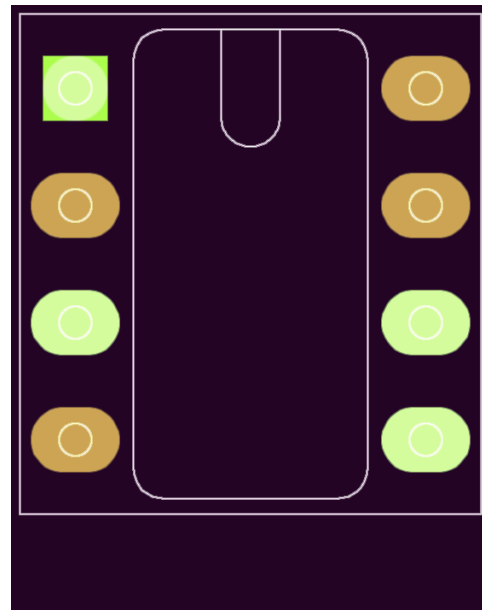

*Fig 12. Part in **FreePCB***

*Fig 13. Part in **Freerouter***

Footprint graphics are copied but line widths are limited to one pixel. Also **Freerouter** has no text import facility so text and Reference designators are not included in the design file.

Rules and Clearance Classes

FpcROUTE uses a number of *FreePCB* default and DRC values to define the boards initial setting and default rule set. This rule set also define the "Default" and "Outlines" clearance classes.

FreePCB Source Parameter	*Freerouter* Settings
Route snap_angle	snap_angle
Project default via pad	DefVia

default rule set
{

Project default trace width	(trace) width

Implied Default Clearance Class
{

DRC trace to trace	<default clearance>
DRC pad to trace	smd_to_turn_gap
DRC Cu area to Cu area	default_area
DRC pad to pad	via_via
DRC pad to pad	via_smd
DRC pad to pad	via_pin
DRC pad to pad	pin_pin
DRC pad to pad	smd_pin
DRC pad to pad	smd_smd

}

Implied Outlines Clearance Class
{

DRC Bd edge to any Cu	default_Outlines
DRC Bd edge to any Cu	area_Outlines
DRC Bd edge to any Cu	via_Outlines
DRC Bd edge to any Cu	smd_Outlines
DRC Bd edge to any Cu	pin_Outlines
(set to 0.00)	Outlines_Outlines

}
}

To change a DRC value in FreePCB, start a DRC Check from the context or Tools menu, make the desired parameter value(s) and run the DRC by hitting the Check button. Hitting the Close button will <u>not</u> save any new values.

Net Classes

In *FreePCB*, nets can be assigned non-default trace widths and/or via sizes. *FpcROUTE* will group any non-default nets into net classes with matching widths and via sizes so that *Freerouter* can use the non-default values.
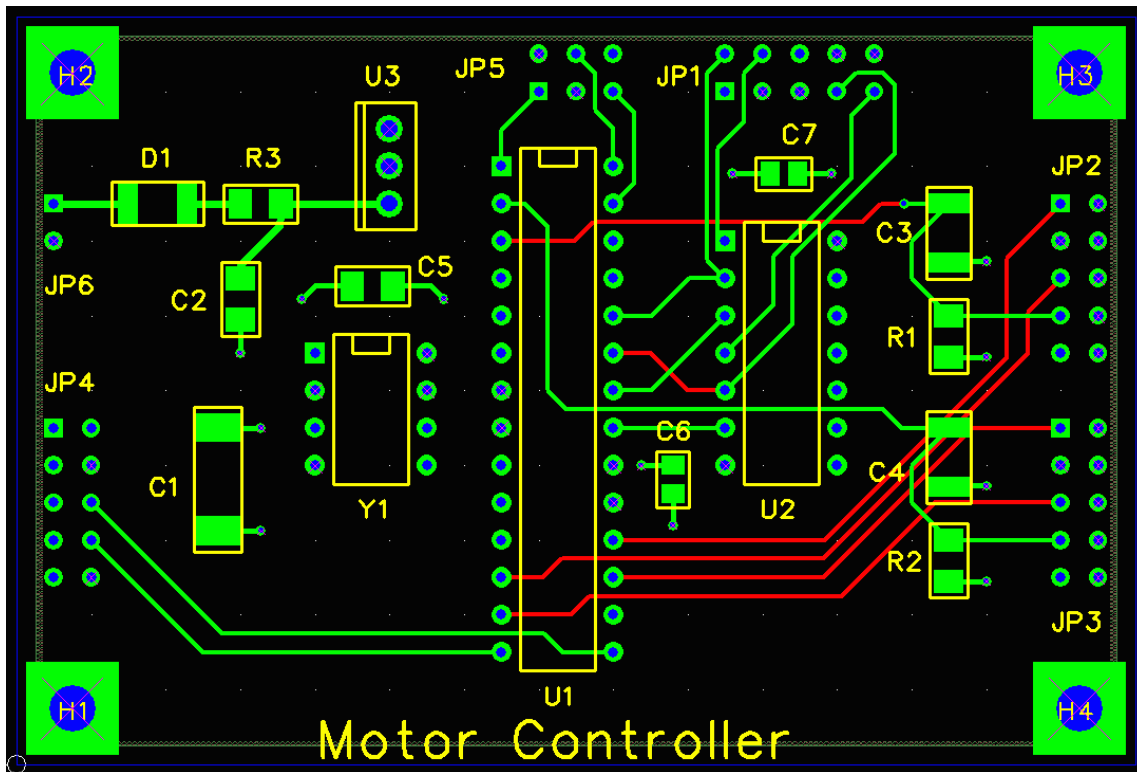
Examples
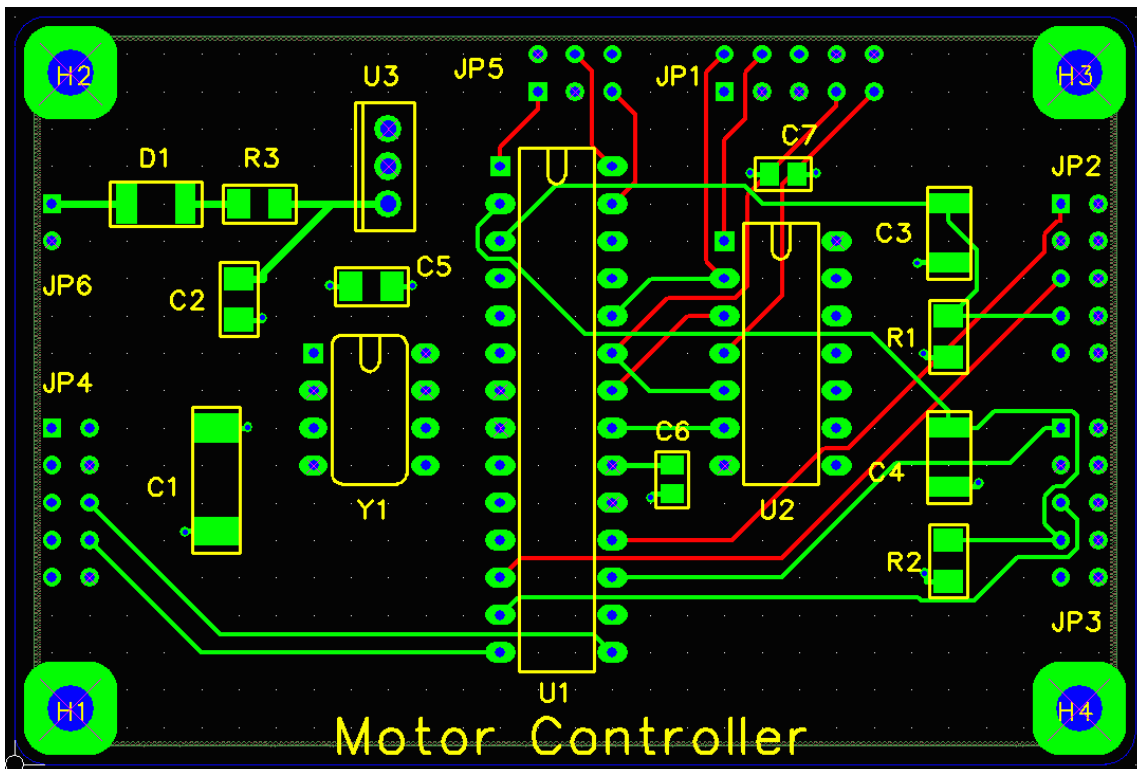

Fig 14. Hand routed by Allan Wright

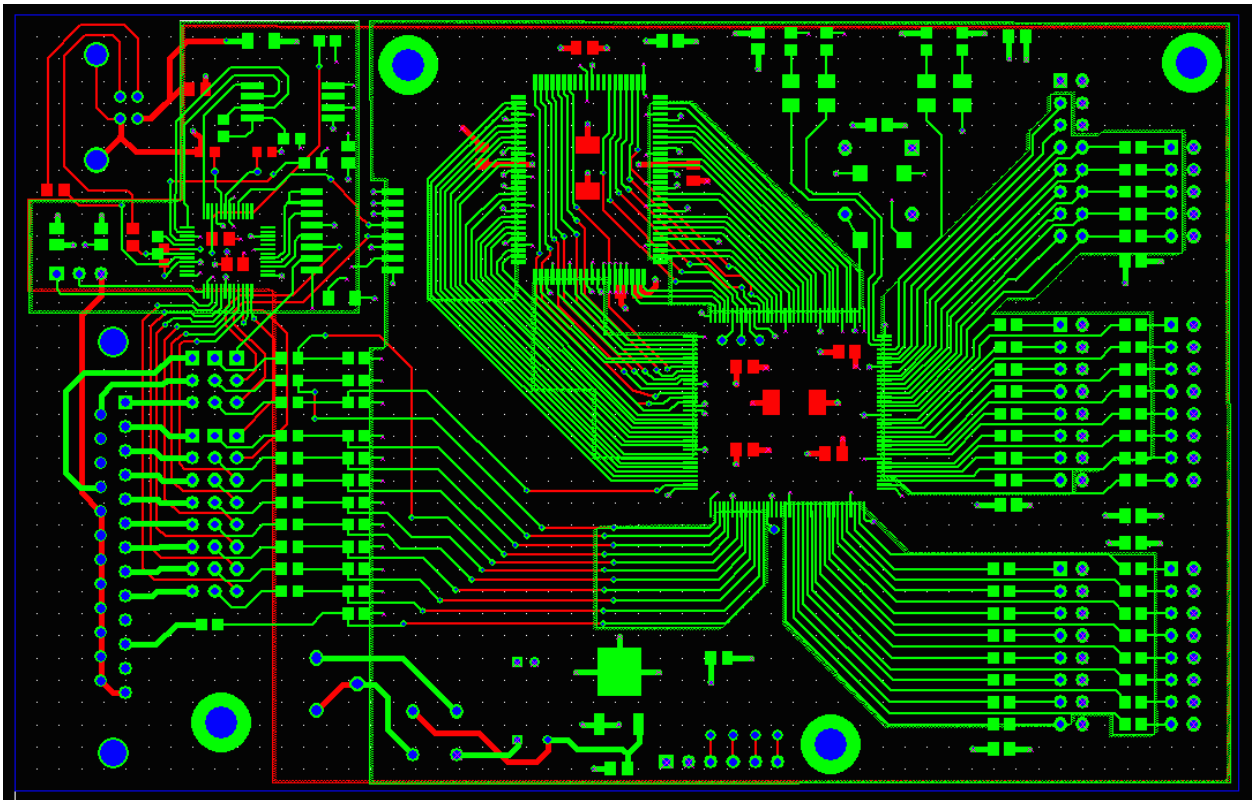*Fig 15. Autorouted by Freerouter*

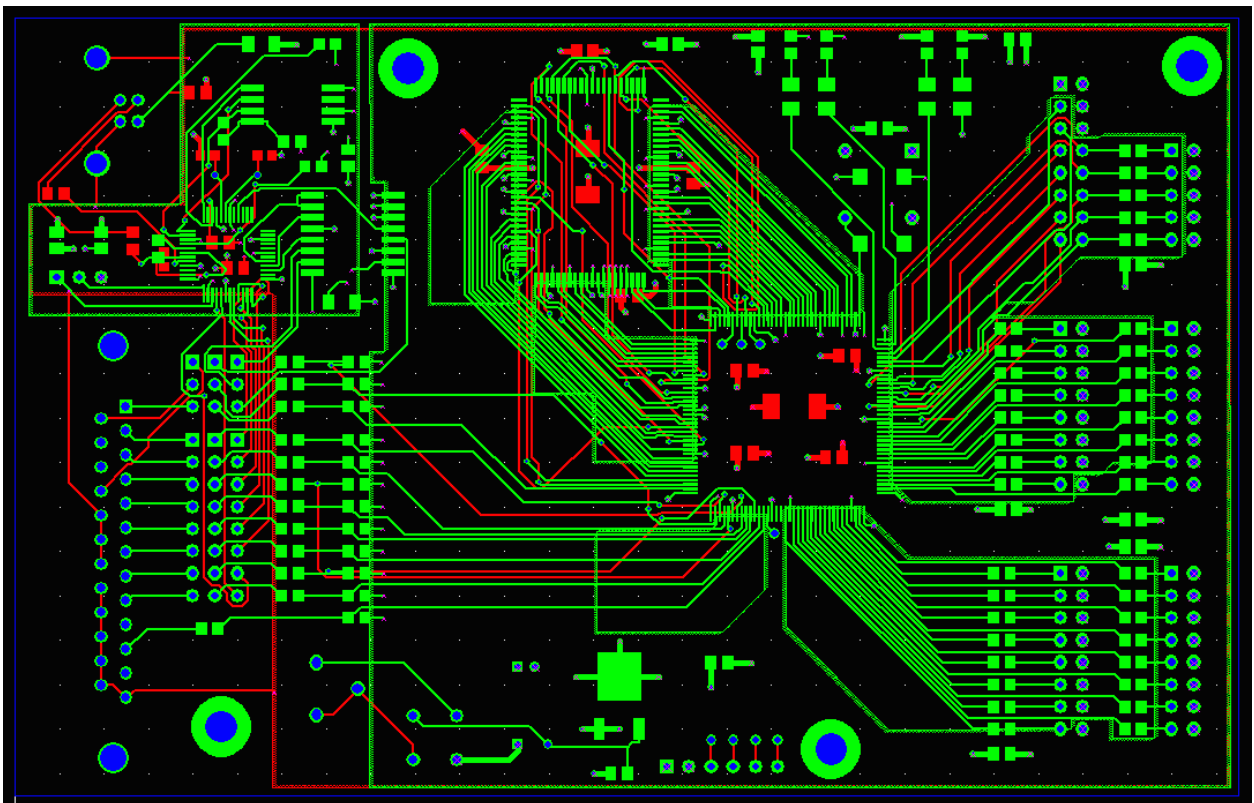*Fig 16. Hand routed by Bob Grieb*

Fig 17. Autorouted by Freerouter

Example Comments

The first example, on page 20, uses a board file created by an unknown old version of *FreePCB*. Attempting to load it produced an error:

```
        FpcROUTE - A FreePCB <--> Freerouter file translator
                    Ver 1.100R (c)2007 Bruce Parham

   (motor.fpc - 187) Error 10460: reading layer list
```

Examining the file in a text editor revealed that two layer were missing. The missing (soldermask cutouts) layers made the first copper layer number 10, not the expected 12.

```
n_copper_layers: 4
  layer_info: "selection" 0 255 255 255 1
  layer_info: "background" 1 0 0 0 1
  layer_info: "visible grid" 2 255 255 255 1
  layer_info: "highlight" 3 255 255 255 1
  layer_info: "DRC error" 4 255 128 64 1
  layer_info: "board outline" 5 0 0 255 1
  layer_info: "rat line" 6 255 0 255 1
  layer_info: "top silk" 7 255 255 0 1
  layer_info: "bottom silk" 8 255 192 192 1
  layer_info: "thru pad" 9 0 0 255 1
  layer_info: "top copper" 10 0 255 0 1
  layer_info: "bottom copper" 11 255 0 0 1
  layer_info: "inner 1" 12 64 128 64 1
  layer_info: "inner 2" 13 128 64 64 1
```

A warning message was added to alert users of possibly incompatible file versions. The error still exists but now the user has a clue why it fails to load.

```
        FpcROUTE - A FreePCB <--> Freerouter file translator
                    Ver 1.100R (c)2007 Bruce Parham

  Warning: Loading FreePCB file format version <Unknown>.
          This may cause problems...

   (motor.fpc - 187) Error 10460: reading layer list
```

Loading the file and saving it with the current version of FreePCB fixed the file version problem but revealed another error:

```
        FpcROUTE - A FreePCB <--> Freerouter file translator
                    Ver 1.100R (c)2007 Bruce Parham

     File "motor.fpc" loaded, 1262 lines read.

   Error 40587: Part Y1 has unknown pin: B1
```

The board file loaded successfully but the design file generator found a bogus pin. In the text editor, a search for "B1" uncovered the fact that net N01846 wanted to connect pin 9 of U1 to pin B1 of Y1.

```
net: "N01846" 2 0 0 0 0 0 1
  pin: 1 U1.9
  pin: 2 Y1.B1
```

A quick look at component Y1 shows that it uses footprint 8DIP300:

```
part: Y1
  ref_text: 1270000 177800 270 11176000 3048000
  package: "8DIP300"
  shape: "8DIP300"
  pos: 20320000 27940000 0 90 0
```

And a look at the footprint shows pins called 1, 2, 3, ... 8 but no pin B1.

```
name: "8DIP300"
author: "Ivex"
source: "DIGIKEY CATALOG NO. 941, PAGE 64"
  units: MIL
  sel_rect: -58 -35 358 335
  ref_text: 50 -100 150 270 7
  outline_polyline: 7 -50 50
    next_corner: 350 50 0
    next_corner: 350 250 0
    next_corner: -50 250 0
    close_polyline: 0
  outline_polyline: 7 -50 100
    next_corner: 0 100 0
    next_corner: 0 200 0
    next_corner: -50 200 0
    close_polyline: 0
  n_pins: 8
    pin: "1" 28 0 0 0
      top_pad: 2 55 27 27 0
      inner_pad: 1 55 27 27 0
      bottom_pad: 2 55 27 27 0
    pin: "2" 28 100 0 0
      top_pad: 1 55 27 27 0
      inner_pad: 1 55 27 27 0
      bottom_pad: 1 55 27 27 0
    pin: "3" 28 200 0 0
      top_pad: 1 55 27 27 0
      inner_pad: 1 55 27 27 0
      bottom_pad: 1 55 27 27 0
    pin: "4" 28 300 0 0
      top_pad: 1 55 27 27 0
      inner_pad: 1 55 27 27 0
      bottom_pad: 1 55 27 27 0
    pin: "5" 28 300 300 0
      top_pad: 1 55 27 27 0
      inner_pad: 1 55 27 27 0
      bottom_pad: 1 55 27 27 0
    pin: "6" 28 200 300 0
      top_pad: 1 55 27 27 0
      inner_pad: 1 55 27 27 0
      bottom_pad: 1 55 27 27 0
    pin: "7" 28 100 300 0
      top_pad: 1 55 27 27 0
      inner_pad: 1 55 27 27 0
      bottom_pad: 1 55 27 27 0
    pin: "8" 28 0 300 0
      top_pad: 1 55 27 27 0
      inner_pad: 1 55 27 27 0
      bottom_pad: 1 55 27 27 0
```

The final fix was to delete the whole net with the text editor. Must have been a bum netlist...

The second example, Fig 16 on page 21, is quite a challenge because it uses only two layers with traces embedded in copper areas. Notice how Bob carefully routed the trace group, in the lower left quadrant, using traces on the back side to prevent the top copper area from becoming as series of isolated islands. *Freerouter* missed that point because keepouts were not used to control routing.

Single Sided Boards (A brief outline)

Single sided boards, with jumpers, present a challenge for autorouting. To explore how *Freerouter* would handle a single sided project, the *Motor Controller* demo board was modified to use only copper on the bottom side with jumpers replacing any top side traces.
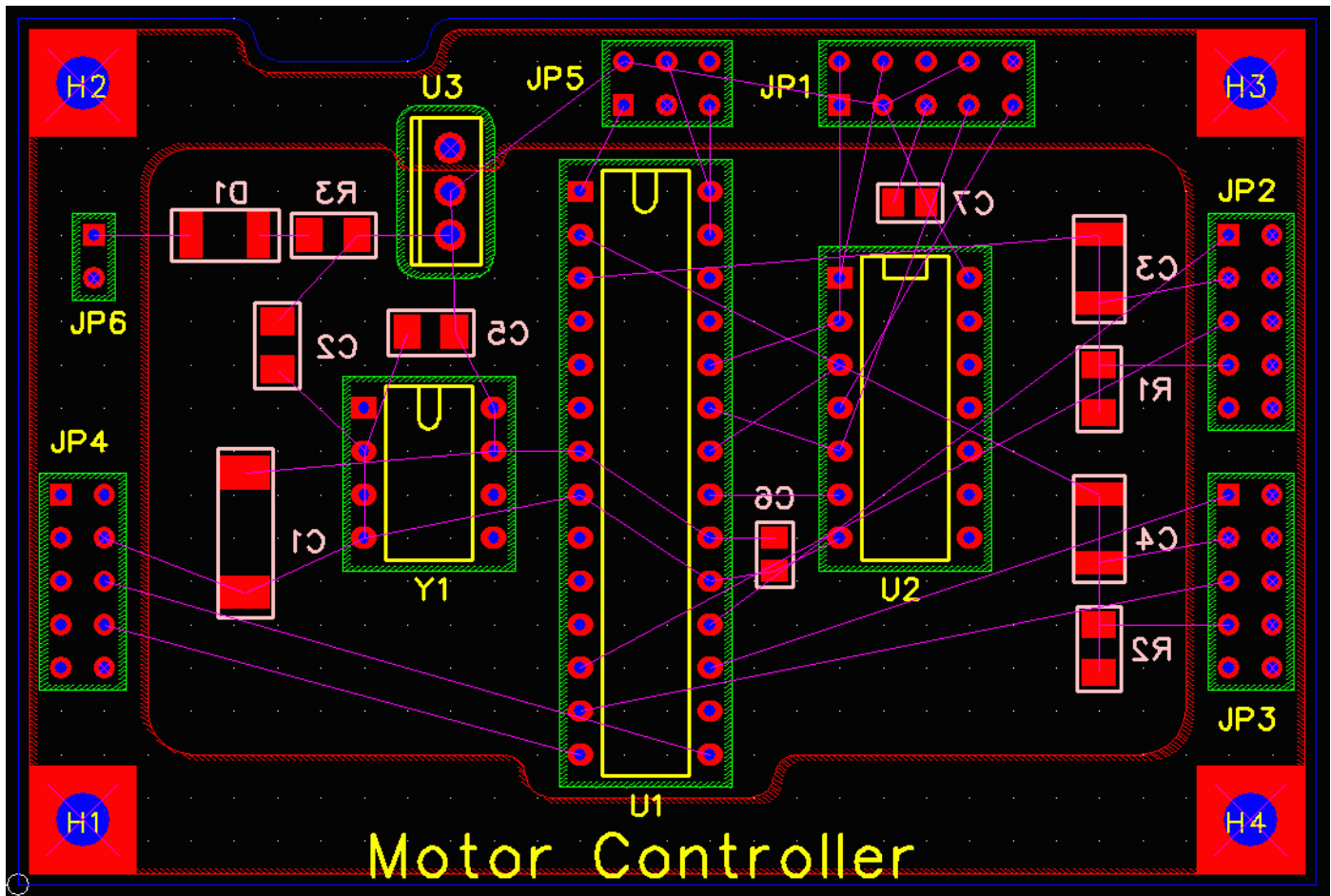
Fig 18. Single Sided Start

Board modifications:

1. All *smt* components were moved to the bottom.

2. *Keepouts* covering all through hole parts were added on the top side to prevent the router from making any direct connections to top side pads.

3. The VCC plane was *deleted*.

4. The GND plane was *moved* to the bottom side and a large *cutout* was added.

5. The DIP footprint *pins* were changed slightly to provide clearance for a 25 mil trace.

6. The *default via* was enlarged to 50 mils diameter.

7. Power trace size was enlarged to 25 mils.

The board file was translated normally and loaded into *Freerouter*.

Fig 19. Ready to route

Before starting, a few Autoroute Parameters were adjusted:

- The layer *Preferred Direction* settings were changed to make the top, jumper, layer vertical.
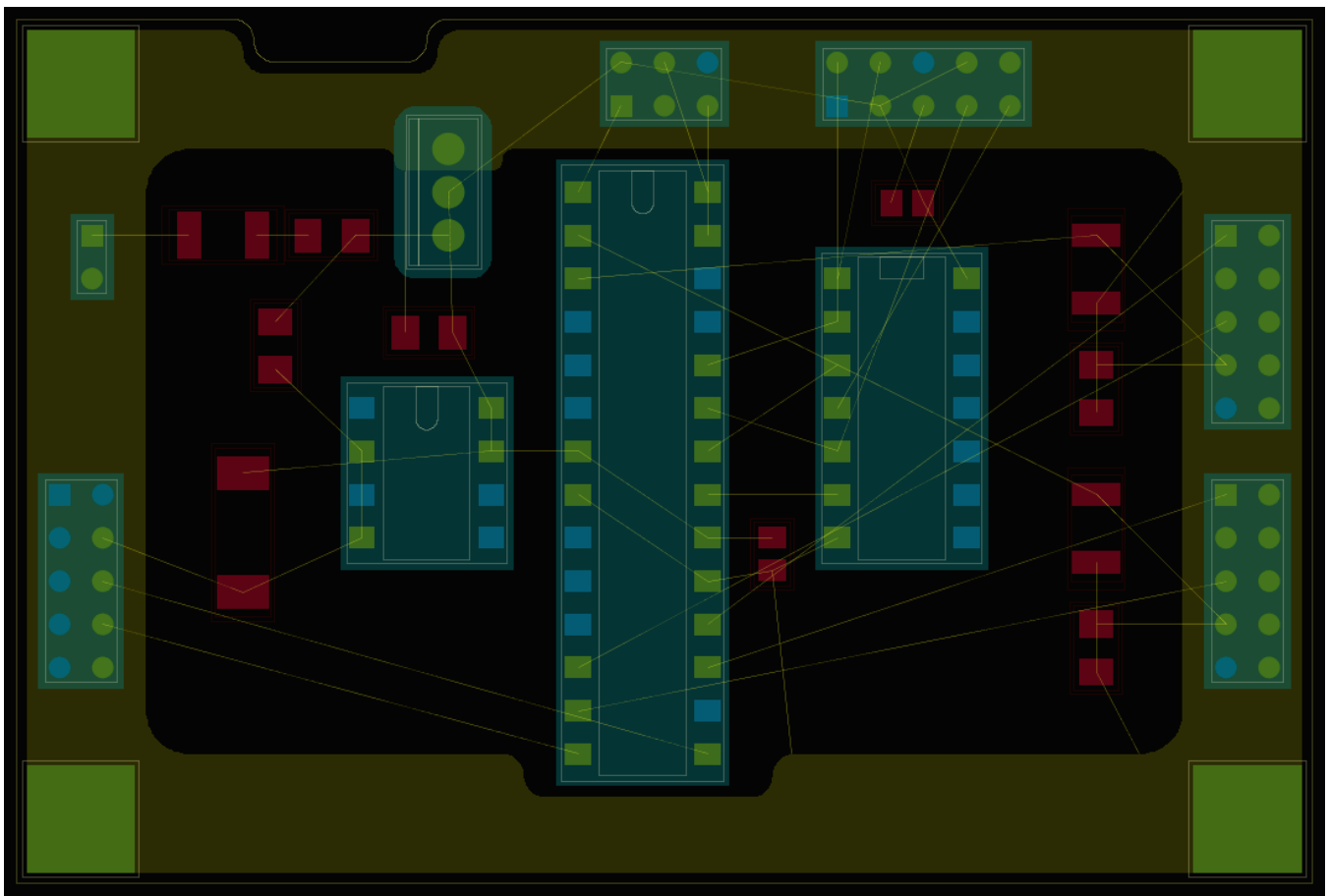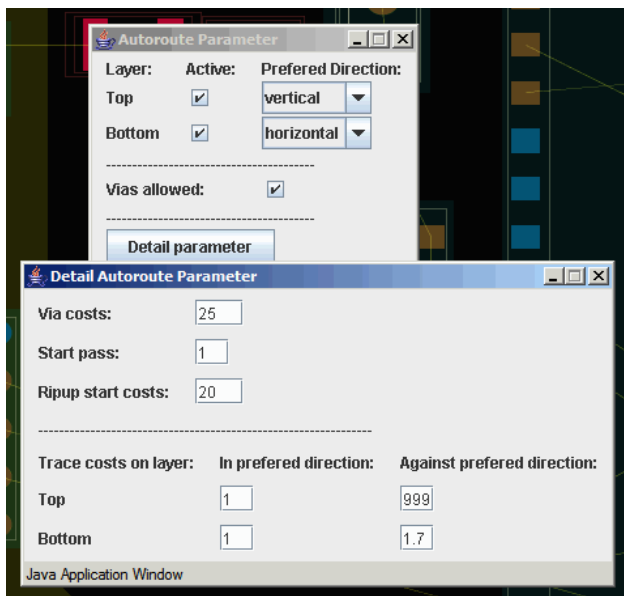
- The *Via* cost was lowered.

- The *Ripup* cost was lowered to encourage more retrys.

- The *Top* side *Against Preferred Direction* cost was set to maximum to force straight jumpers.

Some Clearance Matrix rules also needed changes:

- All Via Rule setting were set to default.

- The trace width for power classes C1 and C2 were set to 25 mils.

Net Classes

name	via rule	clearance class	trace width	on layer	shove fixed	cycles with areas	min. length	max. length
default	default	default	10.0	all	☐	☐	0.0	-1.0
C1	default	default	25.0	all	☐	☐	0.0	-1.0
C2	default	default	25.0	all	☐	☐	0.0	-1.0

Add   Remove   Assign   Select   Show Nets   Filter Incompletes

Java Application Window

At this point, the autorouter was started. In a few minutes the router finished with all connections completed.
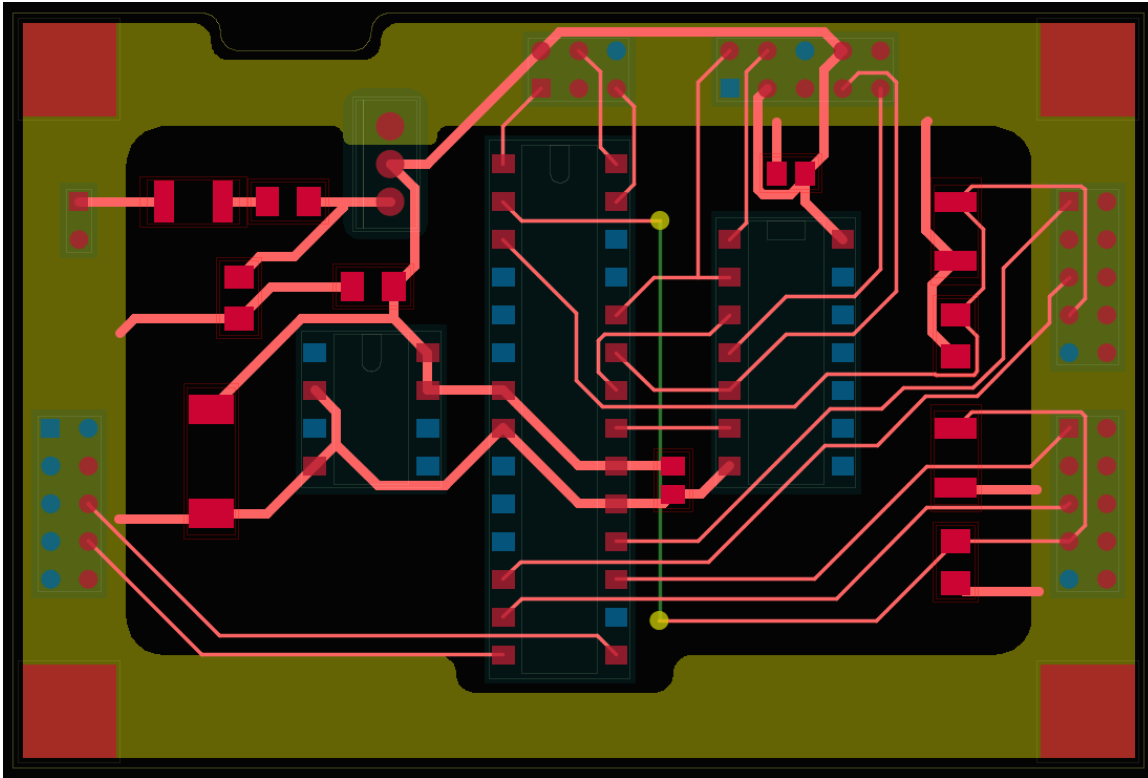
Fig 20. Router finished

Manual cleanup, Fig 21, was done to handle the obvious issues with traces and the ground plane.
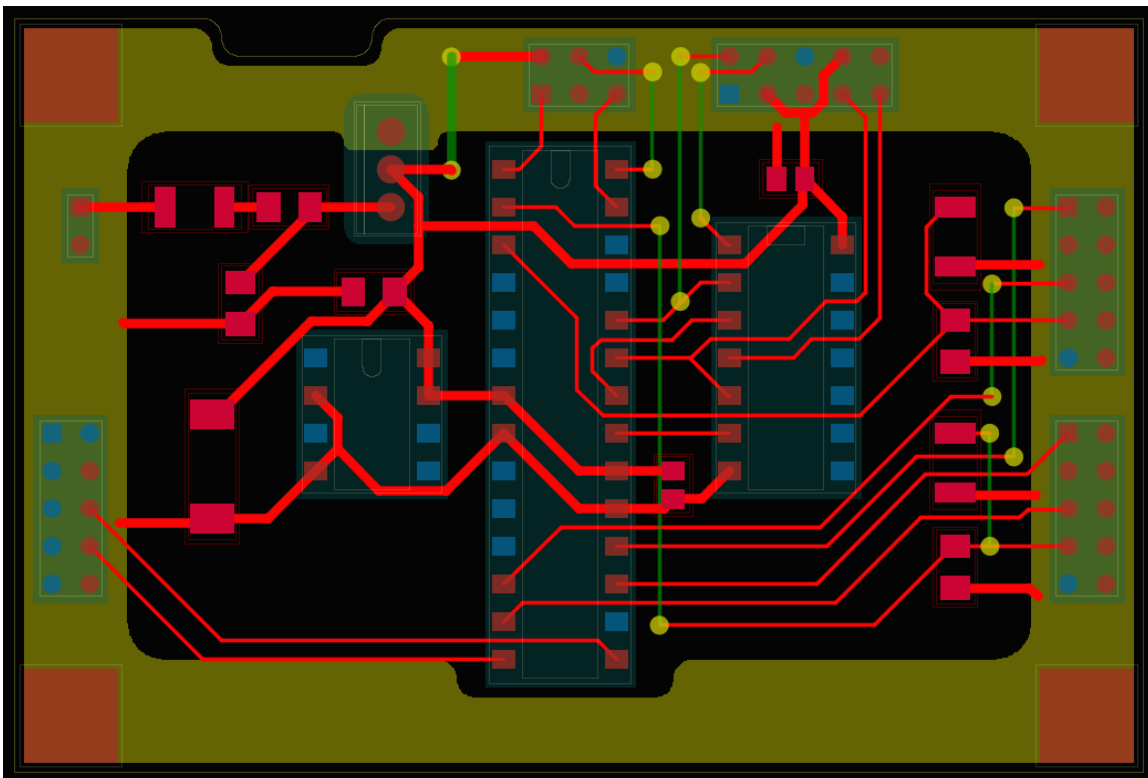
Fig 21. After cleanup

The routed board was imported back into **FreePCB** where a few minor modifications were added. Stub traces were added along the inside edges of the H1-H4 mounting hole pads to cover thermal gaps.
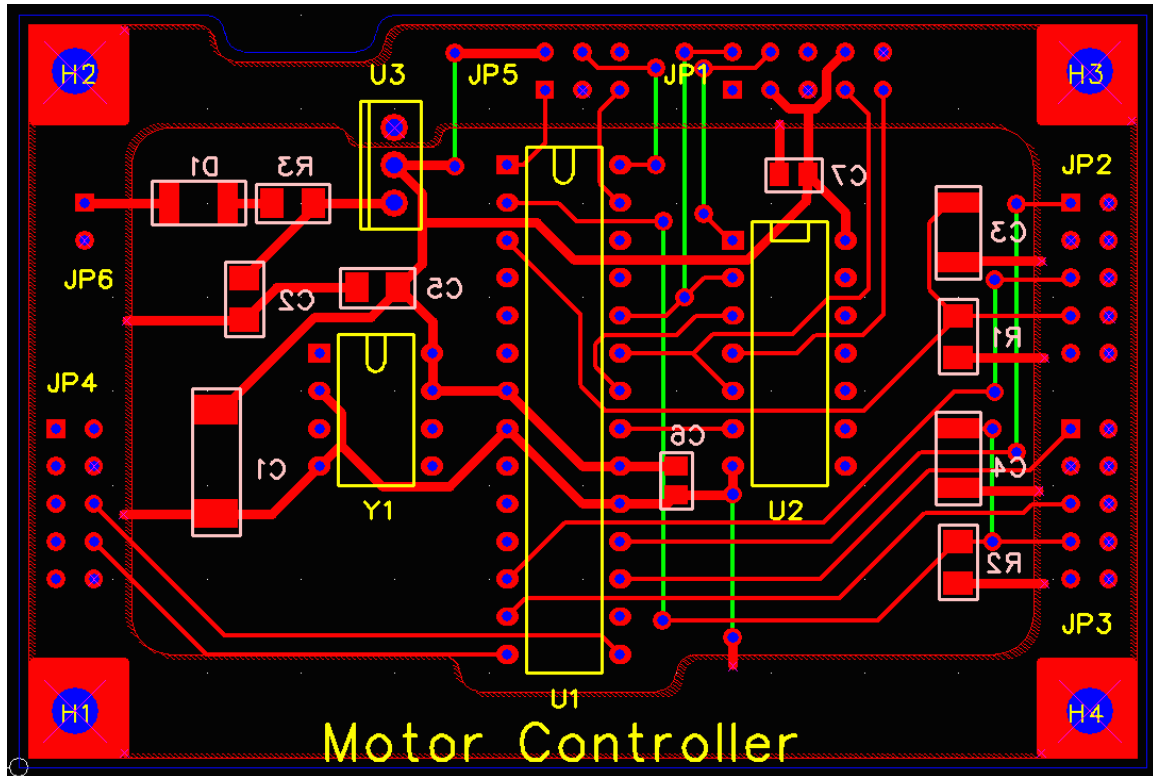
Fig 22. Finished layout

Gerbers were generated with pin thermals and pilot holes enabled and thermals rotated 45º.

Fig 23. Finished Gerbers

Error Codes

FpcROUTE Error Messages (based on Ver 1.101 source code)

The error messages in FpcROUTE are composed of an error code and a text string.
The error code is created by a macro which adds a file dependent offset to the
source code line number where the call to the error display is located.

Most messages are invoked with a fixed string:

```
help(ERR, "Error string");
```

Some use a formatted string:

```
sprintf(buf, "format stuff", value(s));
help(ERR, buf);
```
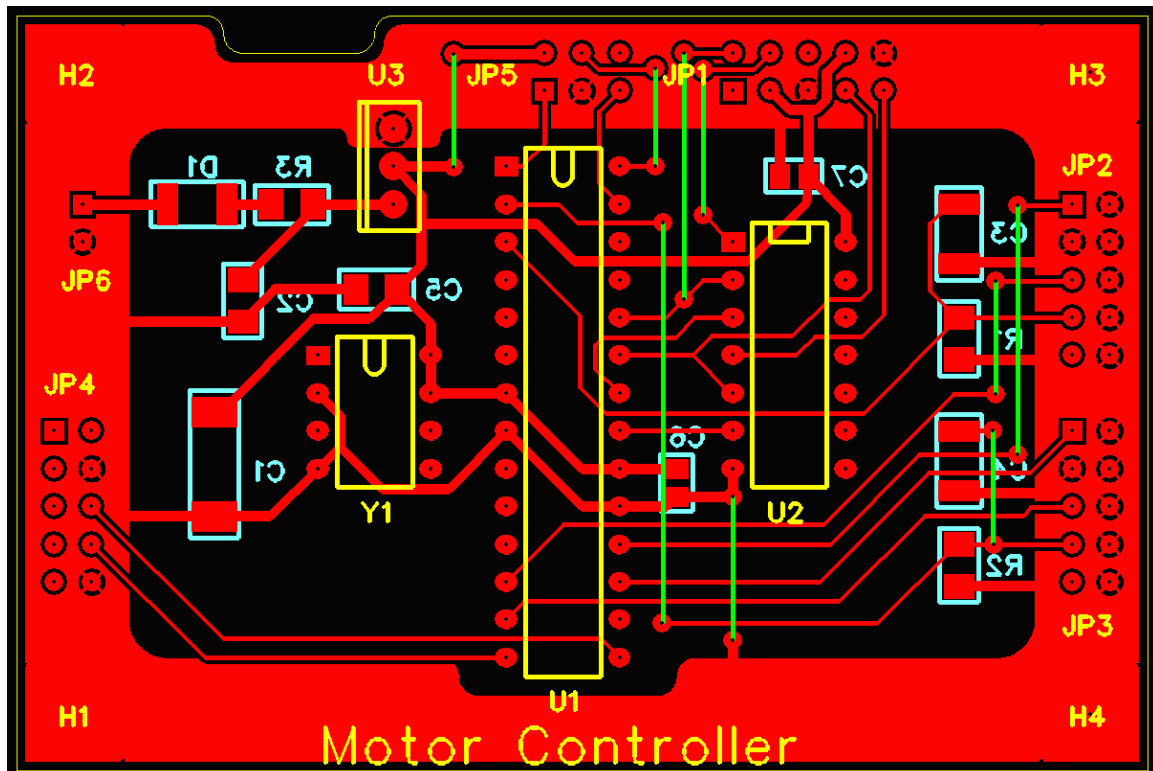
with the macro ERR defined as:

```
#define    ERR    <FILE_OFFSET> + __LINE__
```

The FILE_OFFSET value is unique to each source code file:

```
FpcROUTE.c        0        Startup where main() is located, parse cmd line, top level control
FPC_load.c    10000        FPC file loader - parse file and build initial database
FPC_write.c   20000        FPC file writer - convert database to .fpc file
SES_load.c    30000        SES file loader - parse file, analyze wiring, update database
DSN_write.c   40000        DSN file writer - convert database to .dsn file
PolyProc.c    60000        Generic polygon routines: arcs, inside/outside, etc.
```

Increments of 10000 were used because some of the files contain over 1000 lines of source. In fact,
the session file loader is over 2000 lines because of the code needed to analyze wiring connectivity.

Routines that read and parse input files parenthetically add the file name and line number to the
front of the error string:

```
(filename.fpc - 1234) Error 10456: <error message text>
```

The error numbers listed below are based on version 1.101 source code. The error codes are based on source
code line numbers. As the code evolves, the numbers will chance. User data errors are highlighted in red.

The errors fall into five general categories with the first three being the most common:

```
File reading and parsing - missing record, bad or not enough parameters in a record, wrong version
File creation or opening - bad file name, path or disk full
Database errors/inconsistencies – netlist errors and missing parts
Coding errors/inconsistencies - code debugging
Memory allocation failure - calloc() or realloc() op fails (big board on small laptop?)
```

Error Messages by Source File

FpcROUTE.c

```
                            Text
    ----------------------------------------------------------------------
 1.    Error 105: -U switch needs at least one parameter
 2.    Error 106: -U switch: Zero (0) not a valid outline
 3.    Error 119: Invalid switch xxxx
 4.    Error 126: Too few arguments
 5.    Error 132: Unable to load FreePCB file
 6.    Error 156: Unable to load router session file
 7.    Error 160: Unable to create updated FPC file
 8.    Error 164: Unable to load rules file
 9.    Error 165: Unable to create DSN file
```

FPC_load.c

Text

```
 --------------------------------------------------------------------------
10.   (<fpcfile> - <line#>) Error 10114: allocating grid storage
11.   (<fpcfile> - <line#>) Error 10117: bad grid item record
12.   (<fpcfile> - <line#>) Error 10260: reallocating padstack storage
13.   (<fpcfile> - <line#>) Error 10421: allocating width menu storage
14.   (<fpcfile> - <line#>) Error 10426: premature EOF reading width menu
15.   (<fpcfile> - <line#>) Error 10431: bad width menu item record
16.   (<fpcfile> - <line#>) Error 10437: reading width list
17.   (<fpcfile> - <line#>) Error 10443: bad n_copper_layers: record
18.   (<fpcfile> - <line#>) Error 10446: not enough copper layers, 2 or more needed for autorouting
19.   (<fpcfile> - <line#>) Error 10450: premature EOF
20.   (<fpcfile> - <line#>) Error 10455: bad layer_info: record
21.   (<fpcfile> - <line#>) Error 10458: bad layer_info: record
22.   (<fpcfile> - <line#>) Error 10460: reading layer list
23.   (<fpcfile> - <line#>) Error 10464: allocating Layer Info storage
24.   (<fpcfile> - <line#>) Error 10469: premature EOF reading FPC options
25.   (<fpcfile> - <line#>) Error 10475: allocating padstack storage
26.   (<fpcfile> - <line#>) Error 10513: premature EOF looking for FPC footprints
27.   (<fpcfile> - <line#>) Error 10536: allocating footprint storage
28.   (<fpcfile> - <line#>) Error 10557: missed footprint in pass 2
29.   (<fpcfile> - <line#>) Error 10573: unknown footprint units
30.   (<fpcfile> - <line#>) Error 10579: bad footprint rect
31.   (<fpcfile> - <line#>) Error 10592: bad footprint ref
32.   (<fpcfile> - <line#>) Error 10602: allocating footprint text space
33.   (<fpcfile> - <line#>) Error 10613: bad footprint text
34.   (<fpcfile> - <line#>) Error 10617: bad footprint text record
35.   (<fpcfile> - <line#>) Error 10635: footprint poly space
36.   (<fpcfile> - <line#>) Error 10641: bad polyline header
37.   (<fpcfile> - <line#>) Error 10659: EOF in footprint poly
38.   (<fpcfile> - <line#>) Error 10662: footprint poly space
39.   (<fpcfile> - <line#>) Error 10668: reading footprint poly
40.   (<fpcfile> - <line#>) Error 10673: bad fp poly next_corner: record
41.   (<fpcfile> - <line#>) Error 10681: bad FP poly close_polyline: record
42.   (<fpcfile> - <line#>) Error 10684: footprint poly scan
43.   (<fpcfile> - <line#>) Error 10696: bad n_pins record
44.   (<fpcfile> - <line#>) Error 10699: allocating FP pin storage
45.   (<fpcfile> - <line#>) Error 10705: EOF reading fp pins
46.   (<fpcfile> - <line#>) Error 10706: reading pins
47.   (<fpcfile> - <line#>) Error 10708: bad pin record
48.   (<fpcfile> - <line#>) Error 10709: bad pin name
49.   (<fpcfile> - <line#>) Error 10712: bad fp pin record
50.   (<fpcfile> - <line#>) Error 10719: bad 1st fp pad record
51.   (<fpcfile> - <line#>) Error 10725: bad fp smt top_pad
52.   (<fpcfile> - <line#>) Error 10736: bad fp smt bottom_pad
53.   (<fpcfile> - <line#>) Error 10742: invalid 1st pad record
54.   (<fpcfile> - <line#>) Error 10747: inner pad record missing
55.   (<fpcfile> - <line#>) Error 10751: bad fp inner_pad
56.   (<fpcfile> - <line#>) Error 10758: bottom pad record missing
57.   (<fpcfile> - <line#>) Error 10762: bad fp th bottom_pad
58.   (<fpcfile> - <line#>) Error 10776: premature EOF reading FPC Footprints
59.   (<fpcfile> - <line#>) Error 10812: premature EOF looking for FPC board outlines
60.   (<fpcfile> - <line#>) Error 10833: premature EOF in board outlines Pass 1
61.   (<fpcfile> - <line#>) Error 10842: allocating outline storage
62.   (<fpcfile> - <line#>) Error 10844: No Board Outline found
63.   (<fpcfile> - <line#>) Error 10845: Limits Outline not found
64.   (<fpcfile> - <line#>) Error 10846: Keepin Outline not found
65.   (<fpcfile> - <line#>) Error 10857: premature EOF in outlines Pass 2
66.   (<fpcfile> - <line#>) Error 10860: bad board outline: record
67.   (<fpcfile> - <line#>) Error 10865: allocating outline corner storage
68.   (<fpcfile> - <line#>) Error 10873: bad board outline corner: record
69.   (<fpcfile> - <line#>) Error 10886: premature EOF in outline corners
70.   (<fpcfile> - <line#>) Error 10893: signal outline exceeds board outline
71.   (<fpcfile> - <line#>) Error 10927: premature EOF looking for FPC sm cutouts
72.   (<fpcfile> - <line#>) Error 10948: premature EOF in sm cutouts Pass 1
73.   (<fpcfile> - <line#>) Error 10957: allocating sm cutout storage
74.   (<fpcfile> - <line#>) Error 10967: premature EOF in SM cutouts Pass 2
75.   (<fpcfile> - <line#>) Error 10970: bad sm_cutout: record
76.   (<fpcfile> - <line#>) Error 10978: allocating sm cutout corner storage
77.   (<fpcfile> - <line#>) Error 10986: bad SM cutout corner: record
78.   (<fpcfile> - <line#>) Error 10991: premature EOF in SM cutout corners
79.   (<fpcfile> - <line#>) Error 11036: unknown part angle processing keepouts
80.   (<fpcfile> - <line#>) Error 11040: allocating circular keepout storage
81.   (<fpcfile> - <line#>) Error 11075: premature EOF looking for FPC parts
82.   (<fpcfile> - <line#>) Error 11096: premature EOF in parts Pass 1
83.   (<fpcfile> - <line#>) Error 11105: allocating parts storage
84.   (<fpcfile> - <line#>) Error 11115: premature EOF reading parts header Pass 2
```

```
 85.    (<fpcfile> - <line#>) Error 11118: bad part: record
 86.    (<fpcfile> - <line#>) Error 11120: premature EOF reading parts record 1, Pass 2
 87.    (<fpcfile> - <line#>) Error 11121: reading parts record 1
 88.    (<fpcfile> - <line#>) Error 11124: bad parts ref_text: record
 89.    (<fpcfile> - <line#>) Error 11131: premature EOF reading parts record 2, Pass 2
 90.    (<fpcfile> - <line#>) Error 11132: reading parts record 2
 91.    (<fpcfile> - <line#>) Error 11134: bad parts package: record
 92.    (<fpcfile> - <line#>) Error 11136: premature EOF reading parts record 3, Pass 2
 93.    (<fpcfile> - <line#>) Error 11137: reading parts record 3
 94.    (<fpcfile> - <line#>) Error 11139: bad parts shape: record
 95.    (<fpcfile> - <line#>) Error 11141: premature EOF reading parts record 4, Pass 2
 96.    (<fpcfile> - <line#>) Error 11142: reading parts record 4
 97.    (<fpcfile> - <line#>) Error 11145: bad parts pos: record
 98.    (<fpcfile> - <line#>) Error 11160: unknown footprint
 99.    (<fpcfile> - <line#>) Error 11185: allocating net Tee storage
100.    (<fpcfile> - <line#>) Error 11200: reallocating net Tee storage
101.    (<fpcfile> - <line#>) Error 11210: Tee X value changed
102.    (<fpcfile> - <line#>) Error 11211: Tee Y value changed
103.    (<fpcfile> - <line#>) Error 11240: allocating via storage
104.    (<fpcfile> - <line#>) Error 11250: reallocating via storage
105.    (<fpcfile> - <line#>) Error 11275: adding Class storage
106.    (<fpcfile> - <line#>) Error 11284: allocating Class net pointer storage
107.    (<fpcfile> - <line#>) Error 11291: reallocating Class net pointer storage
108.    (<fpcfile> - <line#>) Error 11309: allocating poly keepout storage
109.    (<fpcfile> - <line#>) Error 11317: allocating poly keepout point storage
110.    (<fpcfile> - <line#>) Error 11361: allocating layer net info storage
111.    (<fpcfile> - <line#>) Error 11394: allocating Class storage
112.    (<fpcfile> - <line#>) Error 11406: premature EOF looking for FPC nets
113.    (<fpcfile> - <line#>) Error 11427: premature EOF in nets Pass 1
114.    (<fpcfile> - <line#>) Error 11436: allocating net storage
115.    (<fpcfile> - <line#>) Error 11454: premature EOF loading FPC nets
116.    (<fpcfile> - <line#>) Error 11455: finding FPC net
117.    (<fpcfile> - <line#>) Error 11458: net name too long
118.    (<fpcfile> - <line#>) Error 11461: bad net name
119.    (<fpcfile> - <line#>) Error 11467: bad net: record
120.    (<fpcfile> - <line#>) Error 11493: allocating net pin storage
121.    (<fpcfile> - <line#>) Error 11497: premature EOF reading net pins
122.    (<fpcfile> - <line#>) Error 11498: missing net pin record
123.    (<fpcfile> - <line#>) Error 11500: bad net pin: record
124.    (<fpcfile> - <line#>) Error 11509: allocating net trace storage
125.    (<fpcfile> - <line#>) Error 11514: premature EOF reading net traces
126.    (<fpcfile> - <line#>) Error 11515: missing net connect: record
127.    (<fpcfile> - <line#>) Error 11518: bad net connect: record
128.    (<fpcfile> - <line#>) Error 11524: allocating trace segment storage
129.    (<fpcfile> - <line#>) Error 11528: premature EOF reading net trace vtx
130.    (<fpcfile> - <line#>) Error 11529: missing net trace vtx record
131.    (<fpcfile> - <line#>) Error 11533: bad trace vtx: record
132.    (<fpcfile> - <line#>) Error 11549: premature EOF reading net trace seg
133.    (<fpcfile> - <line#>) Error 11550: missing net trace seg record
134.    (<fpcfile> - <line#>) Error 11553: bad trace seg: record
135.    (<fpcfile> - <line#>) Error 11568: allocating net area storage
136.    (<fpcfile> - <line#>) Error 11572: premature EOF reading net areas
137.    (<fpcfile> - <line#>) Error 11573: missing net area record
138.    (<fpcfile> - <line#>) Error 11576: bad net area: record
139.    (<fpcfile> - <line#>) Error 11579: allocating area corner storage
140.    (<fpcfile> - <line#>) Error 11583: premature EOF reading net area corner
141.    (<fpcfile> - <line#>) Error 11584: missing net area corner record
142.    (<fpcfile> - <line#>) Error 11587: bad net area corner: record
143.    (<fpcfile> - <line#>) Error 11623: bad copper area layer
144.    (<fpcfile> - <line#>) Error 11663: premature EOF looking for FPC text
145.    (<fpcfile> - <line#>) Error 11684: premature EOF in text Pass 1
146.    (<fpcfile> - <line#>) Error 11693: allocating text storage
147.    (<fpcfile> - <line#>) Error 11703: premature EOF reading FPC text records
148.    (<fpcfile> - <line#>) Error 11705: copying text record
149.    (<fpcfile> - <line#>) Error 11709: bad texts text: record
150.    (<fpcfile> - <line#>) Error 11734: Unable to open <fpcfile> for input
```

FPC_write.c

```
                        Text
    ------------------------------------------------------------------
151.    Error 20044: unable to open <fpcname> for output
152.    Error 20249: SMT pin without any pad
```

SES_load.c

Text

```
-------------------------------------------------------------------------
153.    (<sesfile> - <line#>) Error 30448: Session (.SES) file name matches FreePCB (.FPC) file name, restart
                                            without file extensions
154.    (<sesfile> - <line#>) Error 30483: premature EOF searching for placement
155.    (<sesfile> - <line#>) Error 30492: parsing session place resolution
156.    (<sesfile> - <line#>) Error 30501: premature EOF searching for place data
157.    (<sesfile> - <line#>) Error 30507: premature EOF reading place data
158.    (<sesfile> - <line#>) Error 30510: reading place data
159.    (<sesfile> - <line#>) Error 30513: parsing quoted place
160.    (<sesfile> - <line#>) Error 30521: locating part
161.    (<sesfile> - <line#>) Error 30530: premature EOF reading place status
162.    (<sesfile> - <line#>) Error 30575: allocating part pin storage
163.    (<sesfile> - <line#>) Error 30601: <refdes>: bad rotation angle (<angle>)
164.    (<sesfile> - <line#>) Error 30613: Footprint: <fp_name>  Pin: <pin_name>  smd padstack invalid
165.    (<sesfile> - <line#>) Error 30633: allocating point storage
166.    (<sesfile> - <line#>) Error 30681: path following recursion overflow
167.    (<sesfile> - <line#>) Error 30852: allocating wiring path storage
168.    (<sesfile> - <line#>) Error 30868: reading wire path header
169.    (<sesfile> - <line#>) Error 30871: parsing wire path header
170.    (<sesfile> - <line#>) Error 30872: bad layer in wire path
171.    (<sesfile> - <line#>) Error 30879: parsing path point record
172.    (<sesfile> - <line#>) Error 30884: adding point to wire path
173.    (<sesfile> - <line#>) Error 30886: premature EOF reading wire path
174.    (<sesfile> - <line#>) Error 30908: allocating wire poly storage
175.    (<sesfile> - <line#>) Error 30916: reading wire polygon header
176.    (<sesfile> - <line#>) Error 30918: parsing wire poly header
177.    (<sesfile> - <line#>) Error 30925: parsing wire poly point
178.    (<sesfile> - <line#>) Error 30932: premature EOF reading wire poly
179.    (<sesfile> - <line#>) Error 30939: reading poly window
180.    (<sesfile> - <line#>) Error 30940: premature EOF reading poly window header
181.    (<sesfile> - <line#>) Error 30941: unknown window shape
182.    (<sesfile> - <line#>) Error 30943: reading poly window header
183.    (<sesfile> - <line#>) Error 30945: poly window layer
184.    (<sesfile> - <line#>) Error 30947: allocating wire poly window storage
185.    (<sesfile> - <line#>) Error 30961: parsing wire poly window point
186.    (<sesfile> - <line#>) Error 30968: premature EOF reading wire poly window
187.    (<sesfile> - <line#>) Error 30986: allocating wire via storage
188.    (<sesfile> - <line#>) Error 30991: parsing wiring via
189.    (<sesfile> - <line#>) Error 30995: unknown via padstack
190.    (<sesfile> - <line#>) Error 31017: allocating wiring tee storage
191.    (<sesfile> - <line#>) Error 31084: allocating routed net list storage
192.    (<sesfile> - <line#>) Error 31109: allocating Routed Area storage
193.    (<sesfile> - <line#>) Error 31125: routed area corner storage
194.    (<sesfile> - <line#>) Error 31154: allocating Power Plane Area storage
195.    (<sesfile> - <line#>) Error 31164: allocating routed net trace storage
196.    (<sesfile> - <line#>) Error 31195: allocating routed trace segment storage
197.    (<sesfile> - <line#>) Error 31326: premature EOF searching for routes section
198.    (<sesfile> - <line#>) Error 31333: parsing session route resolution
199.    (<sesfile> - <line#>) Error 31339: premature EOF searching for routes resolution
200.    (<sesfile> - <line#>) Error 31355: premature EOF searching for parser block
201.    (<sesfile> - <line#>) Error 31363: premature EOF searching for network_out section
202.    (<sesfile> - <line#>) Error 31379: allocating WireList storage
203.    (<sesfile> - <line#>) Error 31392: parsing quoted net name
204.    (<sesfile> - <line#>) Error 31396: parsing net name
205.    (<sesfile> - <line#>) Error 31408: EOF reading wire
206.    (<sesfile> - <line#>) Error 31438: unknown wire type
207.    (<sesfile> - <line#>) Error 31626: start/start VIA mismatch 1
208.    (<sesfile> - <line#>) Error 31631: start/start TEE mismatch
209.    (<sesfile> - <line#>) Error 31635: start/start VIA mismatch 2
210.    (<sesfile> - <line#>) Error 31647: start/start VIA mismatch 3
211.    (<sesfile> - <line#>) Error 31659: start/start VIA mismatch 4
212.    (<sesfile> - <line#>) Error 31662: logical error in TEE sorting 1
213.    (<sesfile> - <line#>) Error 31685: start/end VIA mismatch 1
214.    (<sesfile> - <line#>) Error 31690: start/end TEE mismatch
215.    (<sesfile> - <line#>) Error 31694: start/end VIA mismatch 2
216.    (<sesfile> - <line#>) Error 31706: start/end VIA mismatch 3
217.    (<sesfile> - <line#>) Error 31718: start/end VIA mismatch 4
218.    (<sesfile> - <line#>) Error 31721: logical error in TEE sorting 2
219.    (<sesfile> - <line#>) Error 31748: end/start VIA mismatch 1
220.    (<sesfile> - <line#>) Error 31753: end/start TEE mismatch
221.    (<sesfile> - <line#>) Error 31757: end/start VIA mismatch 2
222.    (<sesfile> - <line#>) Error 31769: end/start VIA mismatch 3
223.    (<sesfile> - <line#>) Error 31781: end/start VIA mismatch 4
224.    (<sesfile> - <line#>) Error 31784: logical error in TEE sorting 3
225.    (<sesfile> - <line#>) Error 31806: end/end VIA mismatch 1
226.    (<sesfile> - <line#>) Error 31811: end/end TEE mismatch
```

```
227.    (<sesfile> - <line#>) Error 31815: end/end VIA mismatch 2
228.    (<sesfile> - <line#>) Error 31827: end/end VIA mismatch 3
229.    (<sesfile> - <line#>) Error 31839: end/end VIA mismatch 4
230.    (<sesfile> - <line#>) Error 31842: logical error in TEE sorting 4
231.    (<sesfile> - <line#>) Error 31873: allocating path sequence storage
232.    (<sesfile> - <line#>) Error 31892: allocating path set storage
233.    (<sesfile> - <line#>) Error 32033: unable to open SES file for input
```

DSN_write.c
```
                            Text
        ----------------------------------------------------------------------
234.    Error 40056: Design (.DSN) file name matches FreePCB (.FPC) file name, restart without file extensions
235.    Error 40068: hdr start indent level
236.    Error 40082: hdr end indent level
237.    Error 40114: structure start indent level
238.    Error 40330: structure end indent level
239.    Error 40358: placement start indent level
240.    Error 40399: placement end indent level
241.    Error 40411: library start indent level
242.    Error 40539: library end indent level
243.    Error 40563: unknown RefDes: <refdes>
244.    Error 40576: Part <refdes> has unknown footprint: <footprint>
245.    Error 40587: Part <refdes> has unknown pin: <pin_name>
246.    Error 40606: network start indent level
247.    Error 40626: bad net node
248.    Error 40700: network end indent level
249.    Error 40718: allocating copper trace storage
250.    Error 40779: wiring start indent level
251.    Error 40859: wiring end indent level
252.    Error 40873: unable to open "<dsn_name>" for output
```

PolyProc.c
```
                            Text
        ----------------------------------------------------------------------
253.    Error 60386: allocating poly point storage
254.    Error 60405: allocating poly point storage
```

Known Bugs and Issues

1. Back Annotation Trace Topology

   Problem: Although *FpcROUTE* creates electrically correct trace/via topologies during command line back annotation, the resulting trace/branch geometries are not always valid for *FreePCB*.

   Work Around: Use the *FreePCB* *Import .ses file* function for back annotation. Allan has added additional connectivity analysis to this function that corrects invalid geometries.

Useful Links

Software
> *FreePCB* layout software: http://www.freepcb.com/
> *Freerouter* autorouter software: http://www.freerouting.net/

Gerber Viewers
> *GC-Prevue* free viewer: http://www.graphicode.com/
> *ViewMate* free viewer: http://www.pentalogix.com/

Reference
> *SPECCTRA® Design Language Reference* manual: http://www.eltm.ru/store/spdlr.pdf

Revision History

V1.100	27 Jan 07	Initial release to FreePCB forum. Release version executable and user guide.
V1.101	31 Jan 07	Fixed a bug in Session file placement quoted refdes parsing. Found by Crodan.
V1.102	4 Feb 07	Fixed padstack size/shape matching bug in design file output. Found by 60 Hertz Jig
V1.103	20 Feb 07	Footprint graphics added to library images. All copper area, outline and footprint polygon/polyline arcs approximated. All copper areas set to "System Fixed" and recovered from original fpc file on back annotation. User guide updated.
V1.201	8 Apr 07	Drill graphics added to library images. Added polygon generated shapes to padstacks for oval, rounded rectangle and octagonal pads. Added keepouts at copperless holes to library images. User guide updated.
V1.203	29 Apr 07	Session file net name parsing bug fixed (thanks JLV). If found in source fpc file, paste mask and panelizing parameters added to board database and included in routed fpc file.
V1.204	2 May 07	Router parameters added to routed project (fpc) file.
V1.210	19 Nov 07	FromTo routing options added. FreePCB file support updated to version 1.339. User guide updated.
V1.211	21 Nov 07	FromTo command parsing bug fixed. User Guide: DSN export graphic updated, added FromTo example, minor edits.
V1.300	2 Aug 08	Inner pad keepouts removed from pins with outer copper. FreePCB file support updated to version 1.344. Added untested FreePCB version caution message. Elapsed time resolution increased to milliseconds. Padstack info display updated and expanded.

To Do List

Software is <u>never</u> finished and that certainly includes *FpcROUTE.*

1. Add support for older and newer *FreePCB* file formats. The current version of *FpcROUTE* only supports *FreePCB* file versions 1.312 and newer. Attempting to load an older version will result in a warning message and the load most likely will fail with and error.

2. Bugs? What bugs? My code don't have no stinkin' bugs! How can I fix what doesn't exist? (I wish!)