## What RenumKiCadPCB Does

This program automatically renumbers a KiCad PCB according to various rules then back-annotates the renumbering scheme into the schematic hierarchy. This makes finding components on the PCB very easy. I recommend renumbering early in the layout process because the “copper pour” problem (see below) only shows up after updating. Also, RenumKiCadPCB will spot errors early. I have stress tested with https://github.com/OLIMEX/DIY-LAPTOP/tree/master/HARDWARE/A64-TERES/TERES-PCB1-A64-MAIN_Rev_C which is the most complex KiCad project I have come across. The Rev C files themselves have issues such as ERC and DRC errors, an excess #PWR flag, and there are blank reference designations (i.e. "") on the PCB. All relevant files, including this one, can be found at https://github.com/BrianAtDocumentedDesigns/RenumKiCadPCB0400.

## Installation Notes

There are build instructions below for Windows Visual Studio 2019, Windows Eclipse, Windows MSYS2, and Linux below. I have provided a Windows setup file (RenumKiCadPCB.msi) if that is all you need. If you get a “DLL not found” message, try running and installing vc_redist.x86.exe (https://support.microsoft.com/en-ca/help/2977003/the-latest-supported-visual-c-downloads), which has the distributable DLLs. Chances are they are already on your PC.

## Known Bug

Due to a bug in KiCad PCBNew ([Bug 1609401] Re: PCBnew fails to properly import netlist after changing annotation with pours see https://bugs.launchpad.net/kicad/+bug/1609401). If RenumKiCadPCB is run on a project with copper pours and a schematic netlist is created and imported into PCBNew errors can occur in the netlist import and subsequent DRC. This happens even without using RenumKiCadPCB: sometimes after changing reference designations on PCB and eeSchema if you then generate a netlist and re-import it into PCBNew there may be netlist errors and DRC errors if the board has copper pours.

The fix is simple. Type 'B' and the pours will regenerate except where there are errors where they appear hatched. Note the ratsnest netlist name in the hatched error zone. Edit the zone (hit 'E' near a zone edge) and change the pour net to the ratsnet netlist name. (the ratsnest name is usually near the top of the selection window). Type 'B' again, and the pour will fill. Run DRC and the errors are gone.

## Build Instructions

RenumKiCadPCB is written in c++ and uses the wxWidgets toolkit. To build you have to install wxWidgets on your platform (see https://wiki.wxwidgets.org/Install). If you are on Windows, by far the easiest way to build is with Microsoft Visual Studio otherwise you have to install MSYS2 and get that all working. Another issue with MSYS2 is that you have to statically link the DLLs if you want to redistribute the code, which leads to a huge executable.

Note that compiling or building wxWidgets generates a huge number of warnings, as does building RenumKiCadPCB. As far as I can tell these are all due to wxWidgets.

Note further in Windows that if you install wxWidgets into any other path than C:\wxWidgets-3.0.4 (i.e. if you install wxWidgets-3.1.1, for example) you have to change the relevant paths to includes and library files in your build command or IDE.

These are the source files which are usually installed in the build directory:

RenumDefines.h
RenumExternals.h
RenumLiterals.h
RenumPrototypes.h
RenumStructures.h
RenumKicadPCBClasses.h
RenumKicadPCB_Base.h

RenumGUI0v0_400.cpp
RenumKiCadPCBClasses.cpp
RenumKicadPCB_Base.cpp
RenumGui.fbp
RenumTitle.ico
resource.h
Resource.rc

## Microsoft Visual Studio 2019

Install wxWidgets as per https://wiki.wxwidgets.org/Microsoft_Visual_C%2B%2B_Guide. In Windows I installed at C: so the path to my includes is C:\wxWidgets-3.0.4\include, C:\wxWidgets-3.0.4\include\msvc,  an the library paths are C:\wxWidgets-3.0.4\ and C:\wxWidgets-3.0.4\lib\vc_lib\. Get the “minimal” sample working under MVS. It should be straightforward because wxWidgets has a **minimal.sln** file with all the right configurations. Create a new solution called RenumKiCadPCB and use file explorer to copy Renum0400.sln (or whatever it is now called) into the solution folder, along with all the source files. You should be able to simply rebuild. If you get errors they are likely because you don’t have the right paths to wxWidgets set up.

## Windows/MSYS2

Install wxWidgets as per https://wiki.wxwidgets.org/Compiling_wxWidgets_with_MSYS-MinGW. You **have to run configure** as per the instructions so the right includes and libraries

In Windows (including MSYS2) I installed at C:\ wxWidgets-3.0.4 so if you do it differently you have to change the paths below.

```
g++ -c -g -O0 -mwindows -mthreads -DHAVE_W32API_H -D__WXMSW__ -D_UNICODE -W -Wall -DNOPCH -Wno-ctor-dtor-privacy "-Ic:\\wxWidgets-3.0.4\\lib\\gcc_lib\\mswud" "-Ic:\\wxWidgets-3.0.4\\include" -O3 -Wall -c -fmessage-length=0 -o "RenumGUI0v0_400.o" "RenumGUI0v0_400.cpp"
```

```
g++ -c -g -O0 -mwindows -mthreads -DHAVE_W32API_H -D__WXMSW__ -D_UNICODE -W -Wall -DNOPCH -Wno-ctor-dtor-privacy "-Ic:\\wxWidgets-3.0.4\\lib\\gcc_lib\\mswud" "-Ic:\\wxWidgets-3.0.4\\include" -O3 -Wall -c -fmessage-length=0 -o "RenumKicadPCB_Base.o" "RenumKicadPCB_Base.cpp"
```

```
g++ -c -g -O0 -mwindows -mthreads -DHAVE_W32API_H -D__WXMSW__ -D_UNICODE -W -Wall -DNOPCH -Wno-ctor-dtor-privacy "-Ic:\\wxWidgets-3.0.4\\lib\\gcc_lib\\mswud" "-Ic:\\wxWidgets-3.0.4\\include" -O3 -Wall -c -fmessage-length=0 -o "RenumKiCadPCBClasses.o" "RenumKiCadPCBClasses.cpp"
```

```
g++ -static -static-libgcc -static-libstdc++ -g -mthreads -mwindows "-LC:\\wxWidgets-3.0.4\\lib\\gcc_lib" -o RenumGUI_0400.exe "RenumGUI0v0_400.o" "RenumKiCadPCBClasses.o" "RenumKicadPCB_Base.o" -lwxmsw30ud_core -lwxbase30ud -lwxtiffd -lwxjpegd -lwxpngd -lwxzlibd -lwxregexud -lwxexpatd -lkernel32 -luser32 -lgdi32 -lcomdlg32 -lwinspool -lwinmm -lshell32 -lcomctl32 -lole32 -loleaut32 -luuid -lrpcrt4 -ladvapi32 -lwsock32 -lwininet
```

## Windows/MSYS2 Eclipse

Although I find Microsoft Visual Studio much easier to use, I started with Eclipse. Since it is a major pain to get a wxWidgets program working under Eclipse I thought I would include the instructions here. You have to install MSYS2 an wxWidgets as per above. Then create a new project (i.e. RenumKicadPCB) and make a simple “Helloworld.cpp”. This creates the file structure. Delete all the files in workspace\RenumKiCadPCB\src and copy all the source files (above) into it. Copy **.cproject** and **.project** into the workspace\RenumKiCadPCB\ directory.

### Linux

These instructions assume you have already installed the developer package (see https://linuxconfig.org/how-to-install-gcc-the-c-compiler-on-ubuntu-18-04-bionic-beaver-linux). Then follow wxWidgets install procedure, https://wiki.wxwidgets.org/Compiling_and_getting_started. I installed to /home/brianp/wxWidgets-3.1.2/ and built in /home/brianp/wxWidgets-3.1.2/release. Your build commands may differ if you chose different locations.

**Make sure to run wx-config**: you need to know the flag and linker settings. If you get different setting from me, you have to change your compile and link instructions (below) accordingly.

When I run wx-config –cxxflags I get

```
-I/usr/local/lib/wx/include/gtk3-unicode-3.1 -I/usr/local/include/wx-3.1 -D_FILE_OFFSET_BITS=64 -DWXUSINGDLL -D__WXGTK__ -pthread
```

When I run wx-config –libs I get.

```
-L/usr/local/lib -pthread   -lwx_gtk3u_xrc-3.1 -lwx_gtk3u_html-3.1 -lwx_gtk3u_qa-3.1 -lwx_gtk3u_core-3.1 -lwx_baseu_xml-3.1 -lwx_baseu_net-3.1 -lwx_baseu-3.1
```

```
g++ -D_FILE_OFFSET_BITS=64 -DWXUSINGDLL -D__WXGTK__ -pthread -I/usr/local/lib/wx/include/gtk3-unicode-3.1 -I/usr/local/include/wx-3.1 -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"src/RenumGUIv0_400.d" -MT"src/RenumGUIv0_400.o" -o "src/RenumGUIv0_400.o" "../src/RenumGUIv0_400.cpp"
```

```
g++ -D_FILE_OFFSET_BITS=64 -DWXUSINGDLL -D__WXGTK__ -pthread -I/usr/local/lib/wx/include/gtk3-unicode-3.1 -I/usr/local/include/wx-3.1 -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"src/RenumKicadPCB_Base.d" -MT"src/RenumKicadPCB_Base.o" -o "src/RenumKicadPCB_Base.o" "../src/RenumKicadPCB_Base.cpp"
```

```
g++ -D_FILE_OFFSET_BITS=64 -DWXUSINGDLL -D__WXGTK__ -pthread -I/usr/local/lib/wx/include/gtk3-unicode-3.1 -I/usr/local/include/wx-3.1 -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"src/RenunmKicadPCBClasses.d" -MT"src/RenunmKicadPCBClasses.o" -o "src/RenunmKicadPCBClasses.o" "../src/RenunmKicadPCBClasses.cpp"
```

```
g++ -pthread -L/usr/local/lib -L/home/brianp/wxWidgets-3.1.2/release/lib -o "RenumKiCadPCB0400"  ./src/RenumGUIv0_400.o ./src/RenumKicadPCB_Base.o ./src/RenunmKicadPCBClasses.o   -lwx_gtk3u_xrc-3.1 -lwxregexu-3.1 -lwx_gtk3u_html-3.1 -lwx_gtk3u_qa-3.1 -lwx_gtk3u_core-3.1 -lwx_baseu_xml-3.1 -lwx_baseu_net-3.1 -lwx_baseu-3.1 -lgtk-3
```

As usual you might have to change the flags to mark the application as an executatble.