

# Circuit Board Designer

Authors: Kenneth Shipley, Joseph Spear, and Jason Rivas

## Abstract:

This program provides a simple UI that allows the user to create a circuit schematic using 10 provided components. The user will be able to edit the schematic via drag/drop, delete, connect mechanisms, and, once satisfied with the schematic, the user can export the schematic to an optimal PCB layout image.

## Technology:

- VS Code editor.
- Python 3
- PyQt (or other) library
- Python PIL (Python Imaging Library)/Pillow (PIL fork)

## Requirements:

### A Circuit Builder:

#### A.1 GUI

A.1.1 Implemented using PyQt or similar GUI library

A.1.2 Canvas/Workspace

A.1.2.1 The workspace will have a canvas that represents the 2D grid which contains points for position data that is needed.

A.1.2.1.1 Infinite/Finite option.

A.1.2.1.1.1 Canvas dynamically change size to enclose only the components that are in view and some amount of padding (space around the outside).

A.1.2.1.2 Zoom in/out

A.1.2.1.2.1 [ctrl] + [+] and [ctrl] + [-]

A.1.2.1.2.2 [ctrl] + [scroll up] and [ctrl] + [scroll down].

A.1.2.1.2.3 Zoom option

A.1.2.1.2.3.1 Zoom relative to mouse

A.1.2.1.2.3.2 Zoom relative to center of canvas

A.1.3 Pallet Wheel Widget/Tools(Buttons)

A.1.3.1 Can be expanded and collapsed

A.1.3.1.1 When the center of the widget is clicked, it can expand to show the tools available or collapse to keep workspace minimal.

- A.1.3.1.1.1 When expanded, each option will also expand with its options in another sub-wheel.
        - A.1.3.1.1.1.1 Sub-wheel is concentric to the main wheel.
        - A.1.3.1.1.2 Option expanded will be highlighted.
- A.1.3.2 Can be dragged around
  - A.1.3.2.1 Can be dragged by pressing and holding while moving mouse
- A.1.3.3 Tools/Functions
  - A.1.3.3.1 Wire
    - A.1.3.3.1.1 Allow the user to draw a line connecting two components together.
      - A.1.3.3.1.1.1 Wires can be colored to differentiate from other wires.
      - A.1.3.3.1.1.2 The pin the user first clicks gets the IDs of the following pins the user clicks on which adds these ids to the list of connections in D.1.1.2.5
  - A.1.3.3.2 Wire Snipper
    - A.1.3.3.2.1 Allow the user to cut a wire that they had previously drawn, removing the connection they made.
      - A.1.3.3.2.1.1 This will remove connections from the data of each component in D.1.1.2.5
  - A.1.3.3.3 Add Component
    - A.1.3.3.3.1 Pop up another menu of components available to add
    - A.1.3.3.3.2 Allow the user to pick which component they want to add, then drag it to the workspace
      - A.1.3.3.3.2.1 Will create a new component with the selected one's type in its "Type" field
        - A.1.3.3.3.2.1.1 Adds it to the list of components
      - A.1.3.3.3.2.2 Will create a sprite for the component selected that will follow the mouse until the user lets go.
  - A.1.3.3.4 Delete Component
    - A.1.3.3.4.1 Allow the user to pick which component they want to remove from their schematic
      - A.1.3.3.4.1.1 Will remove any connections the component had

A.1.3.3.4.1.2 Will remove the component from the list of components used

A.1.3.3.5 Label

A.1.3.3.5.1 Allow user to label a particular component

A.1.3.3.6 Comments

A.1.3.3.6.1 Comments will be used in order to annotate the board.

A.1.4 Hamburger Menu/File Management (Buttons)

A.1.4.1 Save Project

A.1.4.1.1 This will save the project as described in D.1

A.1.4.2 Load Project

A.1.4.2.1 This will load the project as described in D.2

A.1.4.3 Create New Project

A.1.4.3.1 Ask user if they want to save then save or don't save (if current instance is not blank)

A.1.4.3.2 Delete current instance of the schematic class (D.1)

A.1.4.3.3 Instantiate a new schematic

A.2 Database of components (Surface Mount Device only)

A.2.1 Will have 10 types of components that the user can choose from

A.2.1.1 Resistor, Capacitor, Inductor, Bipolar PNP/NPN Transistor, Switch, Diode (rectifying and light emitting), Voltage Source, and Ground.

A.3 Data Management

A.3.1 All components created will create instances of objects with the following fields/member variables, all of which are described in sub-numbers of D.1.1.2

A.3.1.1 Label, Type, ID, Position, Connections, and Paths

A.3.2 All components are added to a list (or dictionary) as they are created, and removed from the list as they are deleted.

## **B Optimization of PCB Layout:**

B.1 Grid System

B.1.1 The layout for the PCB will be made on a grid (3D grid if we get time)

B.1.1.1 All components will be laid on the vertices of the grid

B.1.1.2 All component connections will run along the grid connecting the components together

B.1.1.3 2D vector (or 3D if we have time)

B.2 Monte Carlo

B.2.1 Input Arguments

B.2.1.1 List of Components

- B.2.1.1.1 List of dictionaries described in D.1.1.2
    - B.2.2 Output Data
      - B.2.2.1 List of Components now with path data
        - B.2.2.1.1 List of dictionaries described in D.1.1.2
  - B.3 A-Star (A\*)
    - B.3.1 Input Arguments
      - B.3.1.1 List of components
        - B.3.1.1.1 List of dictionaries described in D.1.1.2
    - B.3.2 Output Data
      - B.3.2.1 List of components with updated path data
        - B.3.2.1.1 List of dictionaries described in D.1.1.2
- C Diagram-to-Image Exporter:
  - C.1 Produce an image in either JPEG/PNG
    - C.1.1 The exporting software will take in the resulting position data from the PCB Optimization.
      - C.1.1.1 Uses PIL to draw rectangles for solder pads and lines for traces
    - C.1.2 The program will save the data using functions described in D.1 before exporting to an image and save the output from B
- D Saving/Loading a Circuit Project:
  - D.1 Save to .obj with Python Pickle
    - D.1.1 This will save the current project to a .obj file in order to load the project if the user decides to close the program and wants to open it back up. It will serialize the data using the built in Pickle library. Ideally saving just a single class of a Schematic.
      - D.1.1.1 Every project will have a base of a Schematic class
      - D.1.1.2 Schematic will be made from Components, components have
        - D.1.1.2.1 Label
          - D.1.1.2.1.1 Name given to a component by the user, only there for the user
        - D.1.1.2.2 Type
          - D.1.1.2.2.1 Type of component as described in A.2.1.1
        - D.1.1.2.3 ID
          - D.1.1.2.3.1 Number given to the component for the program to reference
          - D.1.1.2.3.2 Each of its pins will be: ID\_#
        - D.1.1.2.4 Position
          - D.1.1.2.4.1 Circuit Position

- D.1.1.2.4.1.1 Relative to top-left of screen and corresponds to the center of the sprite drawn
  - D.1.1.2.4.1.2 Component has a location on the workspace showing where the user placed the component
    - D.1.1.2.4.1.2.1 Just for redrawing the circuit when loading
  - D.1.1.2.4.2 PCB Position
    - D.1.1.2.4.2.1 Component has a location on a grid showing where the component is located on the pcb
      - D.1.1.2.4.2.1.1 Relative to top-left of screen and corresponds to the central location of the component's solder pads
    - D.1.1.2.4.2.2 Sub-Positions
      - D.1.1.2.4.2.2.1 Location of the solder pads relative to the PCB position
    - D.1.1.2.4.2.3 Will be null if the circuit has not been converted to a pcb
  - D.1.1.2.5 Connections
    - D.1.1.2.5.1 Contains dictionary of connections where the key is the starting pin and the values are the connecting pin(s)
  - D.1.1.2.6 Paths
    - D.1.1.2.6.1 Starts out null.
    - D.1.1.2.6.2 List of points describing path for the connections described in D.1.1.2.5.1
- D.1.1.3 Schematics will also have comments, comments have
  - D.1.1.3.1 Comment
    - D.1.1.3.1.1 A string for written by the user for the comment
  - D.1.1.3.2 Position
    - D.1.1.3.2.1 Location of the comment on the workspace
- D.2 Load project from .obj
  - D.2.1 This will load in the .obj file that saved a previous Schematic instance
    - D.2.1.1 The loaded schematic object will be assigned to a new instance of the Schematic class

Timeline:

Week of	Joseph	Ken	Jason
2/8	Skeleton of the GUI. Implement the two main classes.		
2/15	Finish GUI skeleton.		
2/22	GUI: Component Designs	Database	A*
3/1	GUI: Component Designs/ Tools	Save and Load	A*
3/8	GUI: Tools / Component Drag and Drop	Save and Load	A*
3/15	GUI: Zoom / Pallet Wheel	Monte Carlo	
3/22	GUI: Pallet Wheel / Pallet Wheel Drag and Drop	Monte Carlo	
3/29	GUI: Pallet Wheel / Linking with Component Designs	Monte Carlo	
4/5	GUI: Griding/ Colors	Diagram Image Converter	
4/12	GUI: Hamburger Menu	Diagram Image Converter	
4/19	GUI: Hamburger Menu / Spruce up designs	Diagram Image Converter	

4/26	Testing and Debugging
5/3	

**\*Anytime a task is completed early, that person will help another partner with their task\***