

1 Macro definition

This code starts by defining the image dimensions (WIDTH and HEIGHT), the number of bits used to represent the number of components (NUM_COMPONENTS_BITS), the size of the image in bytes (IMAGE_BYTES), and the maximum number of components per PCB (MAX_COMPONENT_PER_PCB).

2 Main function

Then function prototypes are provided in this code. The code declares function prototypes for ‘template_mode’, ‘list_mode’, and ‘connection_mode’, which perform specific tasks in each mode. This will allow the main function to call these three functions before the details of these two functions are given. In terms of readability, function prototypes also let viewers know there will be two main codes for this program. The three functions in the main function support three modes of operation: template mode *t*, list mode *l*, and connection mode *c*. The main function processes command-line arguments and dispatches control to specific functions based on the selected mode. In *t* mode (printing templates mode), the main function calls “template_mode”, in *l* (list component mode), the main function calls “list_mode”, in *c* mode (identify connection mode), it calls “connection_mode”.

3 *template_mode* function

The *template_mode* function is responsible for processing a template file, specifically for generating a visual representation of a PCB (Printed Circuit Board) template. Upon execution, this function reads the template file provided as input and performs the following steps:

1. Reads the number of components
2. Extracts binary data for a specified template index
3. Visual representation (It interprets the binary data to display the pattern, where 0s may represent empty spaces, and 1s may indicate specific elements or components.)

4 *list_mode* function

The *list_mode* function enables the comparison of a given PCB image with components from a template file. It aims to identify and report which components from the template are present in the PCB image. The primary functionality of this function can be broken down into the following key steps:

1. Conversion of PCB image to binary data
2. Reads the number of components from the template
3. Matching Component (For each component in the template, the function checks if it matches any region within the PCB image. If a match is found, the function records and prints the details of the component found, such as its type, row position, and column position in the PCB image.)

The *list_mode* function assists in quality control and validation processes by confirming the presence of specific components in a PCB image and providing users with a list of the found components.

5 *connection_mode* function

The primary objective of this function is to determine if components in the PCB image are connected and report any direct connections found. It achieves this through a series of steps:

1. Similar to *list_mode* function
2. Component connectivity detection (Identify which components share connections and provides insights into the circuit’s overall connectivity. If a connection is detected between two or more components, the function records and prints this information, indicating which components are connected.)

5.1 Supporting functions

The code includes several supporting functions for *connection_mode* function working with images, such as creating an empty grid, printing a grid, freeing grid memory, checking cell validity, and performing depth-first search (DFS) for component connections.