

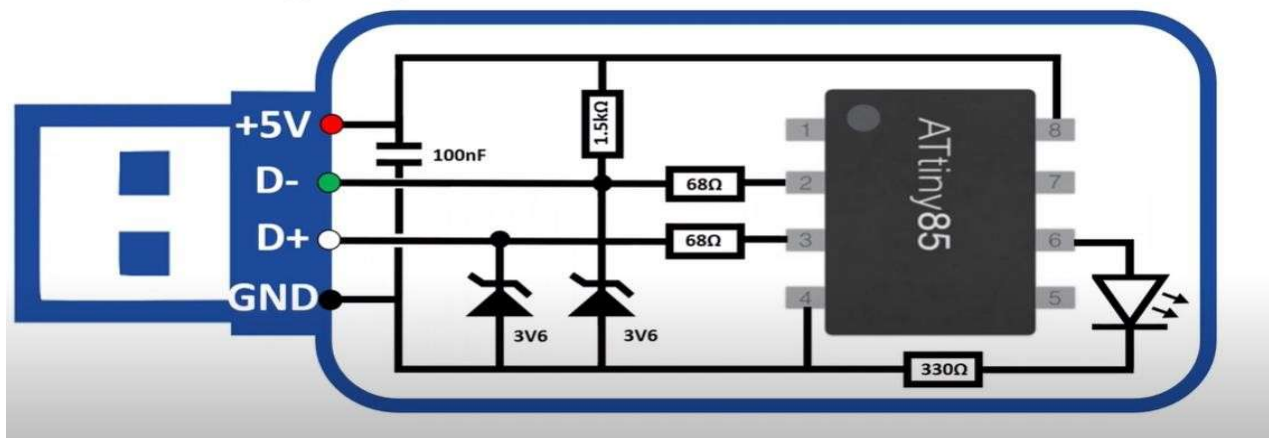
DigiSpark-USB PCB Design using KiCAD

‡ DigiSpark USB:

The DigiSpark USB is a tiny, low-cost microcontroller development board based on the ATtiny85 microcontroller. It's designed for simple projects that need basic USB connectivity and can be programmed via USB using the Arduino IDE. Here's a quick overview of its features and potential applications. **Key Features:**

1. Microcontroller: ATtiny85 with 8 KB of Flash memory, 512 bytes of SRAM, and 512 bytes of EEPROM.
2. USB Support: Direct USB connectivity with a bootloader for programming, eliminating the need for an external programmer.
3. Small Size: Compact, with a size similar to a USB stick, making it great for projects that need to save space.
4. I/O Pins: It has six I/O pins, of which some are shared between USB and general purpose.
5. Power Supply: Operates at 5V when plugged into USB; can also work at 5V or 7-35V from an external source.

The DigiSpark Hardware



Common Applications:

1. USB Automation: Can be programmed to act as a Human Interface Device (HID) like a keyboard or mouse, allowing you to automate repetitive tasks on a computer.
2. Wearable Electronics: Due to its small size, it's ideal for embedding in wearable gadgets or small enclosures.
3. Simple Sensors and Actuators: Controls basic sensors and actuators for small robotics or IoT projects.
4. Hobbyist and Educational Projects: Often used for teaching microcontroller basics due to its simplicity and low cost.

‡ KiCAD:

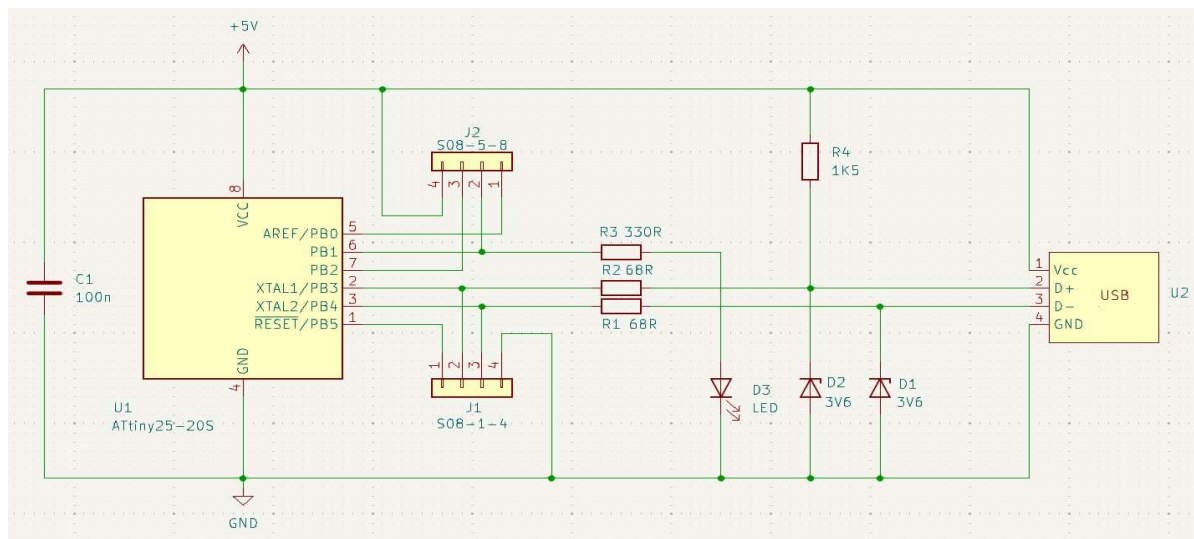
KiCAD is a powerful, open-source software suite for electronic design automation (EDA), particularly well-suited for creating printed circuit boards (PCBs). It's popular among hobbyists and professionals alike because of its comprehensive tools for schematic capture, PCB layout, and 3D visualization. Here's an overview of KiCAD and how it can be beneficial for PCB design projects.

Users can export their designs to Gerber files, which are the standard for PCB manufacturing. KiCAD also generates drill files required for hole placement. The software is compatible with Windows, macOS, and Linux, making it accessible to a broad user base. With its intuitive interface and extensive features, KiCAD is a popular choice for creating anything from simple circuits to complex embedded systems.

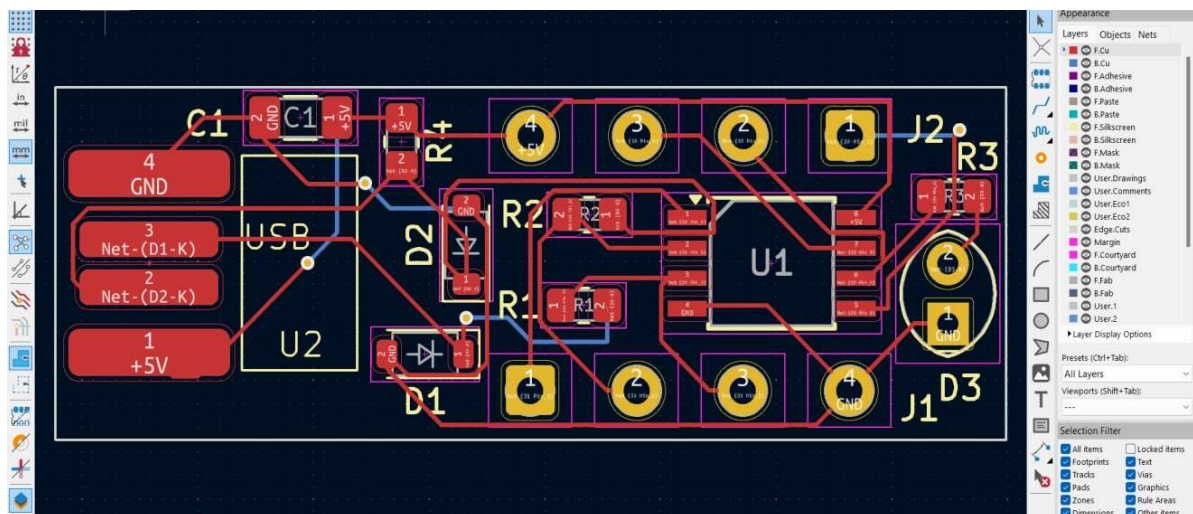
Key Features of KiCAD:

1. **Schematic Capture:** KiCAD provides a schematic editor where you can create the electrical connections for your circuit. It supports hierarchical designs and allows you to add custom symbols.
2. **PCB Layout Editor:** The PCB editor is where you can place components and route connections between them. KiCAD supports multiple layers, including signal, ground, and power planes, making it suitable for complex designs.
3. **3D Viewer:** One of the standout features of KiCAD is its built-in 3D viewer, which gives you a realistic view of your PCB with all components placed. This helps in visualizing the final product and identifying potential issues.
4. **Component Library Management:** KiCAD has extensive libraries for components and footprints, and you can create or import custom libraries for parts not included by default.
5. **Design Rule Checking (DRC):** Ensures that your design meets specified spacing and trace width requirements, reducing the chance of errors in the PCB manufacturing stage.
6. **BOM Generation and Gerber Export:** KiCAD can generate Bill of Materials (BOM) and export files in Gerber format, which are standard for PCB fabrication.

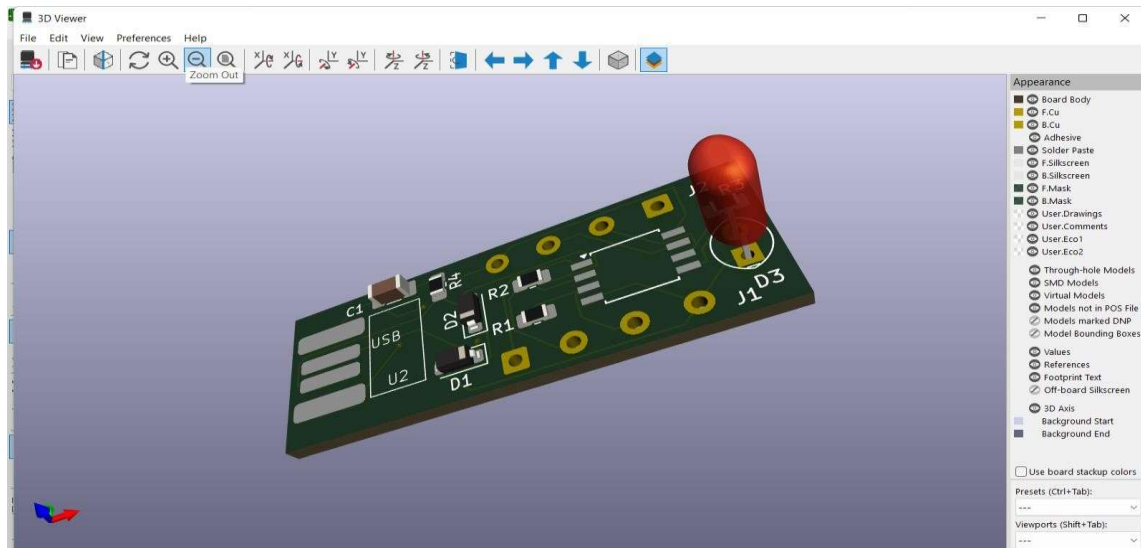
Schematic Drawing:



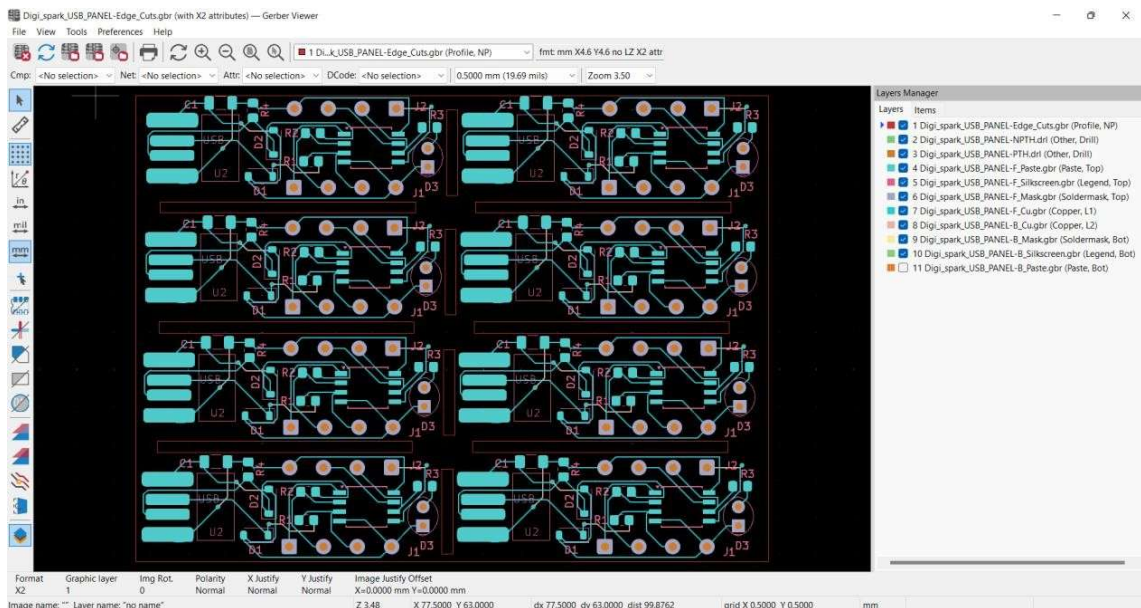
Footprint & PCB Layout:



3D-VIEW of PCB:



Gerber & Drill Files:



✦ **Step-by-step guide to each of these key tasks in KiCAD for designing a PCB & creating custom symbols and footprints, and preparing files for manufacturing:**

1. Schematic Drawing:

- **Open KiCAD Project:** Start a new project in KiCAD.
- **Open Schematic Editor (Eeschema):** This is where you'll design the schematic
- **Place Components:** Use the Add Symbol tool to place components from the default libraries. You can search for components by name.
- **Wire Connections:** Use the Wire tool to draw connections between components. Ensure all connections are correct and logical
- **Annotate:** Use the Annotation tool to assign reference numbers (like R1, C1) to each component.
- **Run ERC:** Use Electrical Rules Check (ERC) to identify potential schematic issues like unconnected pins or voltage level mismatches.

2. Create a Custom Symbol:

- **Open Symbol Editor:** From the main KiCAD window, open the Symbol Editor.
- **Create New Symbol:** Click on File > New Symbol and enter a name. Choose the library where you want to save it (create a custom library if needed).

- Draw Symbol: Use tools like Pin, Rectangle, and Text to design the symbol.
- Add Pins: Place pins for each electrical connection point, ensuring you label each pin with the correct number and name.
- Properties: Set pin properties (e.g., input, output, power) as required.
- Save Symbol: After creating the symbol, save it to the library and use it in your schematic by selecting it in the Add Symbol tool.

3. Create a Custom Footprint:

- Open Footprint Editor: From the main KiCAD window, open the Footprint Editor.
- Create New Footprint: Choose File > New Footprint and enter a name. You can also create a new custom library to save this footprint.
- Define Pads: Place Pads on the editor to represent each pin of your component. Set the pad properties (e.g., shape, size, number) to match the part's specifications.
- Pad Numbering: Ensure pad numbers correspond to the pin numbers in your schematic symbol.
- Add Courtyard and Silkscreen: Draw the component outline using the Courtyard and Silkscreen layers to help with placement and assembly.
- Save Footprint: Save your footprint, and you can now assign it to your custom symbol.

4. PCB Layout:

- Open PCB Layout Editor (Pcbnew): Once your schematic is complete, open the PCB editor.
- Import Netlist/Update PCB from Schematic: Click Update PCB from Schematic to load your components onto the PCB layout.
- Place Components: Drag and place components on the board. Keep related components close for easier routing.
- Define Board Outline: Use the Edge Cuts layer to draw the shape and size of your PCB.
- Route Traces: Use the Route Tracks tool to connect component pads according to the schematic.
- Layer Management: If your board is double-sided, use the Layer Manager to switch between layers.
- Power and Ground Planes: Use Add Filled Zones to create copper planes for power and ground, which helps reduce noise and improves stability.
- Design Rule Check (DRC): Run DRC to catch any issues, such as unconnected traces or clearance violations.

5. Create Multi-PCB Panel:

- Define Individual PCBs: Complete each individual PCB design as usual.
- Combine Designs: You can either copy the individual designs into a new layout for paneling or use a KiCAD plugin like KiKit to automate panelization.
- Add Breakaway Tabs/V-Grooves: Use the Edge Cuts layer to define tabs or V-grooves for breaking the boards apart after manufacturing.
- Check Overall Panel Size: Ensure the panel fits within your manufacturing requirements.

6. Generate Gerber & Drill Files:

- Open Gerber Export Tool: Go to File > Plot in the PCB Layout Editor.
- Set Gerber Options: Choose layers to export (e.g., F.Cu, B.Cu, Edge.Cuts). Enable Use Protel filename extensions if required by your manufacturer.
- Generate Drill Files: Click Generate Drill Files in the Plot dialog to create files for all necessary holes.
- Export Files: KiCAD will generate a set of Gerber files and drill files needed for fabrication. Verify files using GerbView to ensure they look correct.

✚ SUMMARY:

Each step from schematic capture to Gerber generation is essential in the PCB design process. Customizing symbols and footprints enables you to create unique or uncommon components, and panelizing allows multiple PCBs in a single batch. Once complete, you'll have a set of manufacturing-ready files for your PCB!.