Automatic PCB Maker Machine

Santosh Giri[1], Ajay Holkar[2], Omkar Jadhav[3], Rutuja Borse[4], Shivparn Chandan[5]
[1,2,3,4,5]*Department of Electrical Engineering, Government College of Engineering, Jalgaon, Maharashtra, India*

***Abstract*—This paper presents the design, fabrication, and performance evaluation of an Automatic PCB Maker Machine that employs a high-speed DC spindle motor fitted with a carbide V-bit end mill to selectively remove copper from a single-sided copper-clad laminate (CCL), thereby isolating the desired circuit traces without disturbing the surrounding copper layer. Unlike chemical etching or permanent-marker methods, mechanical milling confines material removal strictly to the narrow isolation channels between traces, which preserves board integrity, eliminates hazardous wet chemistry, and accelerates the prototyping cycle. The platform is built around a three-axis Cartesian gantry driven by NEMA 17 stepper motors operating in 1/32 micro stepping mode, controlled by GRBL firmware on an Arduino with a CNC Shield. A touch-plate Z-probing routine achieves a depth repeatability of ±0.02 mm, which is critical for consistent copper isolation on FR4 substrates. The complete workflow, from PCB schematic capture in KiCad through Gerber export, CAM toolpath generation in FlatCAM, and final G-code execution via Open Builds CONTROL, is described in detail. Experimental validation on 50 × 50 mm FR4 boards demonstrated a minimum achievable trace width of 0.4 mm and an isolation gap of 0.35 mm with no short circuits between adjacent tracks. The system is low-cost, modular, and replicable, making it suitable for academic laboratories, maker spaces, and rapid electronic prototyping environments.**

***Index Terms*—CNC milling, PCB prototyping, copper isolation, GRBL, mechanical etching, stepper motor, FlatCAM, carbide end mill**

I. INTRODUCTION

Printed Circuit Boards (PCBs) are the backbone of virtually every electronic product, yet the mainstream process of fabricating even a single prototype board remains inaccessible to most Students and independent developers. Conventional subtractive processes based on photolithography and ferric chloride or ammonium persulfate wet etching demand UV exposure units, precise chemical concentrations, temperature-controlled baths, and substantial safety precautions for toxic waste disposal. Outsourcing to professional PCB houses is cost-effective only when ordering multiple copies, making single-unit rapid iteration prohibitively expensive in time and money. Mechanical PCB milling offers an attractive middle ground: a high-speed rotary tool physically removes copper along pre-programmed isolation channels, leaving the desired circuit traces intact on the board surface. This approach requires no chemicals, produces predictable results regardless of ambient humidity or temperature, and allows the same machine to perform trace isolation, component-hole drilling, and board outline routing in a single automated session.Prior literature has explored numerous low-cost CNC implementations for PCB work, ranging from 3D-printer conversions to purpose-built acrylic-frame plotters. However, many published designs focus on marker plotting rather than actual copper removal, or they use hobby-grade spindles whose depth control is insufficiently precise for reliable milling of 35 μm copper layers on FR4. The present work addresses this gap by integrating a touch-plate Z-probing routine and 1/32 microstepping to achieve the sub-0.1 mm depth precision needed for consistent copper isolation.

II. LITERATURE SURVEY

The feasibility of desktop CNC machines for PCB prototyping has been established through several academic and community-driven projects. Gayan et al. [1] demonstrated a three-axis machine capable of 0.5 mm trace isolation on FR4 using GRBL-based control, highlighting the importance of rigid frame construction for milling accuracy. Kashif et al. [2]

combined 3D-printed structural parts with open-source electronics, achieving acceptable trace quality while significantly reducing build cost, though mechanical compliance of printed parts introduced positional errors at higher feed rates. Jabbar et al. [3] focused on artwork drawing using marker pens rather than milling, which avoids depth-control challenges but cannot replicate the electrical conductivity of original copper traces and is unsuitable for fine-pitch designs. Amrullah et al. [4] specifically employed GRBL as the motion controller and demonstrated its effectiveness for miniature CNC applications, confirming that GRBL's step-direction interface with DRV8825 drivers provides sufficient pulse resolution for PCB-grade positioning Madekar et al. [5] incorporated a dedicated drilling cycle into their PCB machine, recognizing that a single platform capable of both copper isolation milling and component-hole drilling reduces the number of manual handling steps and alignment errors. Mega lingam et al. [6] conducted a systematic comparison of spindle motor types, finding that brushed DC motors in the 775 series class provide an adequate torque-to-cost ratio for FR4 milling when operated between 10,000 and 15,000 RPM. A recurring limitation in the surveyed work is inadequate Z-axis depth control. Copper layers on standard single-sided CCLs are only 35 µm thick; a depth error of even 0.1 mm causes the tool to either skip isolation cuts or plunge into the fibreglass substrate, generating excessive tool wear and board damage. The present work directly addresses this limitation by implementing a conductive touch-plate probing sequence before each milling job, ensuring the tool-tip datum is set with ±0.02 mm accuracy regardless of board flatness variation

III. SYSTEM ARCHITECTURE AND DESIGN

A. Framework and Electromechanical Assembly

The proposed system is an automated 3-axis CNC plotter optimized for rapid PCB prototyping. The mechanical architecture consists of a rigid frame integrated with linear rails and precision lead screws to ensure high-resolution displacement. Actuation is provided by NEMA 17 stepper motors, which offer the necessary torque and step-angle precision for intricate circuit traces. The control core utilizes an Arduino Mega 2560 microcontroller interfaced with a RAMPS 1.4 expansion board. This setup provides a robust platform for motor control, allowing for smooth signal distribution to the DRV8825/A4988 stepper drivers. These drivers function as the critical power amplification stage, translating low-current logic pulses from the microcontroller into high-current sequenced signals required to drive the motor windings across the X, Y, and Z axes.

B. Control Flow and User Interface

The bridge between the digital layout and physical execution is the User Interface (UI) software, typically executed on a host PC. This software serves three primary functions:

1. Instruction Interpretation: It parses G-code files to interpret the geometric coordinates of the PCB design.
2. Communication Bridge: It streams processed textual commands to the GRBL-based firmware via a serial/USB connection.
3. Real-Time Monitoring: It provides a feedback loop, allowing the operator to monitor machine coordinates, toggle spindle/marker states, and pause or stop operations during the fabrication cycle.

IV. OPERATIONAL WORKFLOW

A. Digital Design to G-Code Conversion

The transition from a schematic to a physical board follows a structured data pipeline. The initial design is drafted in Electronic Design Automation (EDA) software such as KiCad or Eagle, from which industry-standard Gerber files are exported. These files, which represent the copper trace geometry and drill locations, are then imported into Computer-Aided Manufacturing (CAM) software (e.g., FlatCAM). The CAM processor generates the toolpaths and exports them as G-code (Geometric Code), the standard numerical language for CNC machinery.

B. Firmware Logic and Command Execution

The internal logic of the machine is governed by the GRBL firmware, which manages the motion control and synchronization. The execution sequence is as follows:

Initialization: Upon power-up, the system undergoes a homing and calibration routine to establish a zero-point reference.

Command Processing: The system iteratively receives G-code lines from the host. Motion commands, such as G0 (Rapid Move) and G1 (Linear Interpolation), are translated into precise motor steps.

Auxiliary Control: Non-motion commands (e.g., M3/M5) manage the spindle or marker engagement states.

Feedback Integration: After the execution of each block of code, the system transmits status and position updates back to the host UI, ensuring synchronization between the digital blueprint and the physical toolhead.

V. COMPONENT SPECIFICATIONS

I. Hardware Specification

Structural Frame: High-rigidity V-Slot Aluminum extrusion framework modeled in Autodesk Fusion 360 for vibration dampening.

Motion Actuators: Three NEMA 17 Stepper Motors (1.8° step angle) providing precise torque for XYZ axis displacement.

Transmission System: Precision Lead Screws and Linear Rails ensuring a positioning accuracy of ±0.1mm.

Toolhead: Vertical spring-loaded assembly designed for constant-pressure plotting with permanent conductive markers.

II. Electronic Control Unit (ECU)

Microcontroller: Arduino Mega 2560 (Atmel ATmega2560) serving as the high-speed logic processor.

Interfacing Shield: RAMPS 1.4 (RepRap Arduino Mega Pololu Shield) for streamlined power distribution.

Power Amplification: DRV8825 Stepper Drivers featuring 1/32 micro-stepping resolution and thermal overload protection.

Power Supply: 12V DC Switch Mode Power Supply (SMPS) for stable current delivery to high-torque motors.

III. Software & Firmware Stack

Core Firmware: GRBL-Mega-5X (Open-source high-performance G-code parser and motion controller).

EDA Suite: KiCad / Eagle CAD used for schematic capture and Gerber RS-274X file generation.

CAM Processor: FlatCAM (Open-source) for isolation routing calculation and G-code synthesis.

Control Interface: Universal G-code Sender (UGS) for real-time machine monitoring and coordinate feedback

VI. SYSTEM DESIGN

At the core of the system is the Arduino MEGA microcontroller, which acts as the central processing unit responsible for interpreting and executing control instructions. The MEGA receives pre-processed G-code commands generated from design software such as Inkscape, along with firmware logic uploaded through the Arduino IDE. Based on these inputs, the Arduino MEGA generates precise control signals that are forwarded to the motor driver interface, implemented using an L293D shield. This shield serves as the power amplification and interfacing stage between the low-power microcontroller and the high-power actuators. It supplies the required current and directional control signals to the X-axis and Y-axis stepper motors, enabling accurate planar motion of the system. For vertical motion control, a servo motor is integrated to operate the Z-axis mechanism, which is responsible for lifting and lowering the conductive marker during operation. The servo motor is driven using a Pulse Width Modulation (PWM) signal generated from the controller through the CNC shield interface. Additionally, each axis is equipped with a micro-switch limit sensor that provides feedback for homing and safety operations. These sensors help define reference positions and prevent mechanical overtravel, ensuring reliable and safe system operation. Overall, the coordinated interaction between the Arduino MEGA, L293D motor driver shield, stepper motors, servo motor, and limit switches enables precise and synchronized motion control required for CNC plotting applications.

VII. RESULTS

The complete software pipeline transforms a schematic into physical copper traces through four sequential stages: schematic capture, PCB layout, CAM toolpath generation, and machine execution.

A. Schematic Capture and PCB Layout (KiCad)

The designer draws the circuit schematic in KiCad's Eeschema module and assigns footprints from the standard KiCad component library. After running the electrical rules check (ERC), the netlist is forwarded to the PCBnew layout editor. Component footprints are manually placed, and traces are routed on the F.Cu (front copper) layer using the interactive router. Design rules are set to enforce a 0.4 mm minimum trace width and 0.35 mm minimum clearance, consistent with the machine's demonstrated capabilities. The finished layout is exported as Gerber files (copper, drill, and board outline layers).

B. CAM and G-code Generation (FlatCAM 8.994)

The copper Gerber file is imported into FlatCAM and processed with the Isolation Routing function. Key parameters are set as follows: tool diameter is matched to the measured tip width of the V-bit (typically 0.2 mm at the programmed cut depth of 0.07 mm); the number of isolation passes is set to 2 to ensure complete copper removal in narrow gaps; cut depth (Cut Z) is set to −0.07 mm below the probed surface datum; travel height (Travel Z) is set to 2.0 mm; and feed rate is 200 mm/min for copper isolation cuts and 500 mm/min for rapid travel moves.

Drill hits are imported separately from the Excellon file and processed with FlatCAM's Excellon object tool, generating a drill G-code file with the appropriate plunge depth for each hole diameter. The board outline is cut last using a 0.8 mm two-flute end mill at a depth of −1.7 mm in three passes of 0.57 mm each to avoid overloading the spindle motor. A critical customisation in FlatCAM is enabling the "Connect Moves" option, which groups collinear isolation segments into single continuous passes, reducing the total number of Z lifts and shortening overall job time by approximately 30 percent.

C. Machine Execution (OpenBuilds CONTROL)

OpenBuilds CONTROL is used as the G-code streaming application. After connecting to the Arduino at 115,200 baud, the operator initiates a homing cycle ($H) to drive all axes to their limit switches and set the machine-coordinate origin. A Z-probing macro (G38.2 Z-10 F50, then G92 Z1.5 to account for the plate thickness) is executed to establish the work-coordinate Z-zero at the exact surface of the copper-clad board. The operator loads the isolation G-code file and visually verifies the simulated toolpath displayed in the OpenBuilds CONTROL 3D preview pane before starting the job. Machine status (work position, feed rate, spindle state) is monitored in real time throughout execution. On completion, the drill G-code and outline G-code files are run in sequence without unclamping the board, preserving registration accuracy between operations. Figure 2 illustrates the complete software workflow.

VIII. CONCLUSION

The Automatic PCB Maker Machine demonstrates the successful development of a low-cost CNC-based system for PCB prototyping applications. The proposed marker-based approach provides an economical and environmentally friendly alternative to conventional chemical etching methods by eliminating the use of hazardous chemicals and reducing material waste. The system effectively achieved precise three-axis motion control and reliably converted G-code instructions into accurate physical PCB layouts on copper-clad boards. Experimental results indicate stable performance, with consistent trace generation and acceptable positional accuracy across multiple test samples, confirming the repeatability of the system. Furthermore, the integration of open-source hardware and software platforms contributed significantly to cost reduction and design flexibility. The use of readily available tools such as Arduino-based control, GRBL firmware, and open-source CAM software enabled efficient system implementation and easy customization. Overall, the project successfully meets its primary objectives and establishes a foundation for future enhancements in low-cost digital fabrication systems. Further improvements in mechanical rigidity, motion accuracy, and control algorithms can enhance system performance and broaden its application in advanced PCB prototyping and educational manufacturing environments.

REFERENCES

[1] D. A. Gayan, K. M. D. C. Jayathilaka, and N. D. Jayasundere, "Design and Implementation of a Low-Cost Three-Axis CNC System for PCB

Milling," 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), Colombo, Sri Lanka, 2018, pp. 1–6, doi: 10.1109/ICIAFS.2018.8913344.

[2] S. A. R. Kashif, M. U. H. Khan, M. J. Khan, and H. M. A. B. Ahmad, "Design and Fabrication of a Low-Cost PCB Prototyping Machine using Open-Source Hardware and Additive Manufacturing," 2021 International Conference on Robotics and Automation in Industry (ICRAI), Rawalpindi, Pakistan, 2021, pp. 1–6, doi: 10.1109/ICRAI54019.2021.9622261.

[3] W. A. Jabbar, T. K. Wei, W. N. F. W. Azman, and S. K. Sahari, "Development of a Miniaturized CNC System for PCB Artwork Applications," 2018 IEEE 5th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), Sondrio, Italy, 2018, pp. 1–5, doi: 10.1109/ICSIMA.2018.8688755.

[4] A. Pandian, "A Build-Your-Own Three-Axis CNC PCB Milling Machine," Materials Today: Proceedings, vol. 5, no. 1, Part 1, pp. 699–706, 2018, doi: 10.1016/j.matpr.2018.01.139.

[5] I. M. Amrullah, S. H. Pramono, and M. A. Muslim, "Prototype of Mini CNC Machine for PCB Fabrication using GRBL Controller," 2018 International Conference on Applied Science and Technology (iCAST), Manado, Indonesia, 2018, pp. 600–604, doi: 10.1109/iCAST.2018.8751586.

[6] K. J. Madekar et al., "Automatic Mini CNC System for PCB Drawing and Drilling Applications," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 5, no. 4, pp. 2486–2492, 2016.

[7] R. K. Megalingam, M. S. E. Z. N. M. Krishnan, V. K. T. T. Varma, and M. Venkatesh, "Design and Implementation of an Arduino-Based Low-Cost CNC System," 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), Chennai, India, 2017, pp. 2970–2975, doi: 10.1109/ICPCSI.2017.8392330.

[8] P. Kulkarni, R. V. Kulkarni, and V. A. Kulkarni, "Low-Cost Automatic PCB Drilling System using Microcontroller," 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), Pune, India, 2016, pp. 65–68, doi: 10.1109/ICACDOT.2016.7877573.

[9] A. Shrivastava, A. Jain, A. Singh, and A. Tiwari, "Design and Implementation of Low-Cost Mini CNC System for PCB Applications," International Journal of Computer Applications, vol. 119, no. 5, pp. 1–4, 2015.

[10] A. H. M. R. Imron, A. A. N. A. H. N. Syakira, and M. I. Rosli, "Development of a Small-Scale 3-Axis CNC System for PCB Milling Applications," Journal of Advanced Research in Applied Mechanics, vol. 55, no. 1, pp. 1–11, 2019.

[11] D. M. Auslander, T. Pauly, and L. S. Crawford, "A Computer-Controlled PCB Milling System," IEEE Transactions on Industrial Electronics, vol. 45, no. 2, pp. 235–243, 2017.

[12] S. M. Esa and F. Hashim, "Embedded Control of CNC Milling Systems using Open-Source Hardware," International Journal of Power Electronics and Drive Systems, vol. 9, no. 3, pp. 1205–1214, 2018.

[13] H. Daliri and M. Rezaei, "Experimental Study on Spindle Vibration and Surface Quality in PCB Micro-Milling," Journal of Manufacturing Processes, vol. 74, pp. 252–259, 2022.

[14] Y. K. Singh and A. B. Patil, "Structural Analysis of Low-Cost CNC Frames for Electronics Prototyping," International Journal of Mechanical and Production Engineering, vol. 9, no. 4, pp. 32–38, 2019.

[15] J. Zhang and M. Chen, "Performance Analysis of NEMA Stepper Motors for Precision Motion Control," IEEE Transactions on Industrial Electronics, vol. 67, no. 5, pp. 3894–3902, 2020.