

# Fabrication of Smart CNC Machine Integrated with Wi-Fi Mobile Application and AI-Assisted Command Interface

Dr. D. Kameswara Rao, A.V. Sai Vardhan, C. Venkatesh, G. Akshay, E. Manjunath  
Department of Mechanical Engineering, Mahatma Gandhi Institute of Technology (Autonomous)  
Affiliated to JNTUH | Gandipet, Hyderabad – 500075, Telangana, India

**Abstract** - This paper presents the design and development of a smart mini-CNC machine integrated with a Wi-Fi-based mobile application and an AI-assisted command interface. Conventional CNC machines require complex G-code programming and wired computer systems, limiting accessibility for educational and small-scale applications. The proposed system combines a compact three-axis mechanical structure fabricated using lightweight metal, aluminum, and acrylic components with stepper motors and driver circuits for precise multi-axis motion control. A mobile application developed in Kotlin allows users to send predefined shape commands, control axis movement, and operate the spindle motor wirelessly via an ESP32 Wi-Fi module, eliminating limitations associated with wired communication. Rule-based AI logic embedded in the application interprets user inputs and automatically generates compatible G-code instructions, reducing the need for manual programming. Testing results indicate that the system achieves dimensional accuracy within  $\pm 0.5$  mm, stable wireless communication with latency of 10–20 ms, and consistent repeatability over multiple machining cycles. The system is well-suited for educational use, hobby projects, and small-scale prototyping, and demonstrates the feasibility of integrating mechanical design, embedded electronics, and intelligent software into a cost-effective CNC platform.

**Keywords:** CNC Machine, GRBL Firmware, ESP32, Arduino, Stepper Motor, Mobile Application, G-Code Generation, Wireless Control, AI-Assisted Interface, Smart Manufacturing, Industry 4.0

## 1. INTRODUCTION

Computer Numerical Control (CNC) machines are automated manufacturing systems used to produce components with high precision and repeatability. These machines operate based on programmed instructions, commonly known as G-codes, which define the movement of cutting tools along multiple axes. CNC technology has significantly improved manufacturing efficiency by reducing human intervention, minimizing errors, and enabling the production of complex geometries that are difficult to achieve using conventional machining methods. CNC machines are widely adopted in industries such as automotive, aerospace, electronics, and manufacturing due to their ability to maintain consistent quality and reduce production time.

Despite their advantages, traditional CNC systems are often expensive, bulky, and require skilled operators to write and interpret G-code programs. They also rely on computer-based control systems with wired connections, which increases complexity and limits accessibility — especially for beginners, educational institutions, and small-scale industries. With advancements in embedded systems, wireless communication, and intelligent interfaces, there is a growing demand for compact, low-cost, and user-friendly CNC solutions that do not compromise essential functionality.

This paper presents the design and development of a smart mini-CNC machine that addresses these challenges. The proposed system integrates wireless communication via an ESP32 Wi-Fi module with an intelligent mobile-based control interface developed in Kotlin. Users can interact with the machine using high-level inputs such as predefined shapes and dimension parameters, which are automatically converted into G-code instructions through a rule-based AI logic, thereby eliminating the need for manual programming. The system aims to provide a cost-effective, portable, and accessible CNC solution aligned with smart manufacturing and Industry 4.0 principles.

## 2. LITERATURE REVIEW

Several researchers have contributed significantly to the development of low-cost CNC systems, wireless control interfaces, and intelligent machining platforms. The following review highlights the most relevant prior work that forms the foundation for the proposed system.

- **Patel and Singh (2018)** developed a compact CNC router using Arduino Uno and GRBL firmware (IJERT), demonstrating that acceptable machining accuracy can be achieved with low-cost components for educational applications.

- **Karthik and Ghosh (2019)** explored Bluetooth-based wireless communication between mobile devices and microcontrollers (IJRASET), establishing the feasibility of eliminating wired connections in CNC systems.
- **Dinesh Babu and Nagendra (2020)** implemented a Bluetooth-controlled CNC system (IJARME), validating real-time command transmission without significant delay.
- **Nair and Thomas (2019)** studied GRBL firmware for multi-axis CNC motion control (IJMETMR), confirming its reliability in converting G-code to stepper motor signals.
- **Reddy and Naresh (2021)** designed a three-axis CNC milling prototype (IJIRSE), providing insights into calibration, mechanical alignment, and stepper motor synchronization.
- **Venkatesh and Rao (2022)** introduced Wi-Fi-based CNC control using ESP32 (IJERT), demonstrating reliable wireless communication for Industry 4.0 smart manufacturing applications.
- **Vishnu and Manoj (2023)** developed a mobile application for CNC control (IJETR), emphasizing the importance of user-friendly interfaces in reducing the operational complexity of CNC systems.
- **Kumar and Vardhan (2024)** compared Bluetooth and Wi-Fi communication methods for Arduino-based CNC machines (IRJET), concluding that Wi-Fi via ESP32 provides better range and reliability.
- **Sarma and Lakshman (2025)** discussed modern trends in compact CNC design including IoT integration and wireless control systems (IJETA), aligning with the present system's objectives.

From the literature review, it is observed that while significant progress has been made in wireless and low-cost CNC systems, very few works combine wireless communication, mobile control, and AI-assisted G-code generation in a single integrated platform. The present work addresses this gap.

### 3. SYSTEM ARCHITECTURE AND DESIGN

The proposed system is composed of three tightly integrated subsystems: a mechanical structure, an electronic control unit, and a software-based control interface. Each subsystem is designed for simplicity, affordability, and compatibility with standard open-source tools.

#### 3.1 Mechanical Subsystem

The machine is built as a three-axis (X, Y, Z) CNC system mounted on a rigid metal frame that provides stability and reduces vibration during operation. Motion along each axis is driven by NEMA-17 stepper motors (1.8° step angle, 45 N-cm holding torque) coupled to T8 lead screws, which convert rotational motion into precise linear displacement. Lead screws were chosen over belt drives for better positional accuracy and minimal backlash — critical requirements for precision engraving and cutting operations. Hardened steel guide rods paired with LM8UU linear bearings ensure smooth, low-friction, and well-aligned movement across all axes. A high-speed DC spindle motor rated at 12V and 7000 RPM performs machining operations including engraving and light cutting on soft materials such as MDF, acrylic, foam, and PCB boards. The overall work area is approximately 250 × 250 × 50 mm. The machine bed is made from MDF board with clamping slots, and flexible shaft couplers are used to connect stepper motor shafts to lead screws, absorbing minor misalignments and preventing mechanical stress.

#### 3.2 Electronic Control Subsystem

The electronic subsystem forms the control backbone of the machine. An Arduino Uno microcontroller (ATmega328P, 16 MHz, 5V) loaded with GRBL firmware serves as the core controller, interpreting incoming G-code instructions and generating step-and-direction signals for L2938 stepper motor drivers. Each driver supports up to 1/32 micro-stepping with adjustable current control and built-in over-temperature protection, enabling smooth and precise motor actuation. An ESP32 Wi-Fi module (dual-core, 240 MHz) acts as the wireless communication interface, operating in Access Point mode to receive G-code commands from the mobile application and forward them to the Arduino via UART serial communication. A regulated 12V DC SMPS power supply provides stable voltage to all components. Limit switches on each axis provide homing functionality and over-travel protection, while a normally-closed emergency stop switch instantly halts all operations in case of fault.

**Table 1: Key Components and Specifications**

No.	Component	Model / Specification	Voltage / Current	Purpose
1	Arduino Uno	ATmega328P, GRBL	5V, ~50mA	Main CNC Controller
2	ESP32 Module	WROOM-32, Dual-Core	3.3V, ~500mA	Wi-Fi Communication
3	Stepper Motor	NEMA-17, 1.8°/step	12V, 1.7A/phase	X, Y, Z Axis Motion

No.	Component	Model / Specification	Voltage / Current	Purpose
4	Motor Driver	L2938, 1/32 stepping	8–35V, 2.2A	Motor Signal Amplifier
5	Spindle Motor	DC, 12V, 7000 RPM	12–24V, 2–5A	Cutting / Engraving
6	Power Supply	SMPS, Regulated	12V DC, 5A	System Power
7	Lead Screw	T8, Pitch 2mm	—	Rotary-to-Linear Motion
8	Guide Rods	Hardened Steel Ø8mm	—	Axis Alignment & Support

### 3.3 Software and Mobile Application Subsystem

The mobile application, developed in Kotlin using Android Studio, serves as the primary human-machine interface. Users select predefined geometric shapes (square, rectangle, triangle, circle), specify dimensions, feed rate, cut depth, and spindle RPM, and the application automatically generates GRBL-compatible G-code using an embedded rule-based AI logic. This eliminates the need for manual programming and makes the system accessible to users without prior CNC experience. The ESP32 operates in Access Point mode, allowing the smartphone to connect directly to the CNC system within a range of 20–25 meters without requiring an external network. G-code is transmitted line-by-line via TCP socket communication; the ESP32 forwards each command to the Arduino via UART, and acknowledgment signals ('ok') are relayed back to the app confirming successful execution. Additional app features include real-time X/Y/Z position display, spindle speed and feed rate adjustment sliders, Start/Stop/Pause/Home controls, a G-code text editor for manual customization, operation history log, and a prominently placed emergency stop button.

## 4. METHODOLOGY

The development methodology followed a systematic multi-phase approach to ensure reliable integration of all subsystems.

- **Phase 1 – Requirement Analysis and Conceptual Design:** System objectives were defined based on identified gaps in existing CNC solutions. The three-axis configuration with lead screw motion, ESP32 wireless control, and mobile app interface was finalized.
- **Phase 2 – CAD Modelling:** The mechanical structure was modelled in Fusion 360 to verify dimensions, check component clearances, and finalize the frame design before fabrication, saving time and reducing trial-and-error.
- **Phase 3 – Fabrication and Assembly:** The metal frame was fabricated, guide rods and lead screws were installed, and all mechanical components were assembled with careful alignment to ensure smooth axis motion.
- **Phase 4 – Electronics Integration:** Arduino Uno, L2938 motor drivers, ESP32 module, limit switches, and emergency stop were wired and tested individually before full integration.
- **Phase 5 – Firmware and GRBL Configuration:** GRBL firmware was flashed onto the Arduino and calibrated for steps/mm, feed rate, acceleration, and homing behavior matching the lead screw pitch and motor resolution.
- **Phase 6 – Mobile Application Development:** The Android app was built in Kotlin with a layered architecture (UI, Logic, Communication layers) and tested for shape generation accuracy, Wi-Fi stability, and command responsiveness.
- **Phase 7 – System Integration and Validation:** The complete system was tested first through manual serial G-code input, then through full wireless app-based control, with performance evaluated across multiple machining cycles.

## 5. SYSTEM DATA FLOW

The end-to-end data flow of the system from user input to physical machining follows a well-defined sequential pipeline:

1. **Step 1:** User selects shape and enters parameters (size, speed, depth) in the mobile app
2. **Step 2:** Rule-based AI logic generates GRBL-compatible G-code from user inputs
3. **Step 3:** G-code is transmitted line-by-line from the app to the ESP32 over Wi-Fi (TCP socket)
4. **Step 4:** ESP32 forwards each command to Arduino Uno via UART serial communication
5. **Step 5:** GRBL firmware parses G-code, plans motion trajectories, and controls acceleration/deceleration
6. **Step 6:** Step-and-direction signals are sent to L2938 motor drivers for X, Y, Z stepper actuation
7. **Step 7:** Lead screws convert rotational motion to precise linear movement along each axis
8. **Step 8:** Spindle motor performs cutting/engraving on the workpiece

9. **Step 9:** Acknowledgment signals are relayed back to the app; position data is updated in real time

## 6. RESULTS AND DISCUSSION

The developed mini-CNC system was evaluated across four primary performance domains: G-code generation accuracy, wireless communication reliability, mechanical motion quality, and overall machining performance.

### 6.1 G-Code Generation Results

The mobile application successfully generated accurate GRBL-compatible G-code for all tested shapes (square, rectangle, triangle, circle). No syntax errors were observed in any generated output. Coordinate values correctly reflected user-specified dimensions, and all motion commands (G00, G01, G02, G03) were properly structured. A representative sample G-code output for a 40 mm square profile is shown below:

```
G21 G90 G00 X0 Y0 | M03 S12000 | G01 X40 Y0 F500 | G01 X40 Y40 | G01 X0 Y40 | G01 X0 Y0 | M05 | G00 X0 Y0
```

### 6.2 Wireless Communication Performance

The ESP32 module in Access Point mode established a stable Wi-Fi connection with the mobile application within 1–3 seconds in indoor environments. G-code commands were transmitted line-by-line with no observed data loss or corruption across all test sessions. Average communication latency was measured at 10–20 milliseconds, enabling real-time machine response to control commands (start, stop, pause, homing). No disconnections or buffering issues were observed during continuous multi-shape machining operations, confirming the reliability and suitability of ESP32-based Wi-Fi for real-time CNC control.

### 6.3 Mechanical Motion and Machining Performance

The CNC machine prototype demonstrated stable movement along all three axes with satisfactory positional control after GRBL calibration. Square and circular profiles were successfully machined on foam material with smooth surface finish and minimal tool marks. The average dimensional deviation was within  $\pm 0.5$  mm, indicating moderate precision acceptable for low-load educational and prototyping applications. Stepper motor performance and controller response were stable across repeated test cycles, with no major vibration or backlash issues during normal operation. Repeatability was consistent, with negligible variation in output geometry across multiple runs. Micro-stepping improved motion smoothness, and flexible shaft couplers effectively absorbed minor shaft misalignments.

**Table 2: System Performance Summary**

Parameter	Observed Value	Remarks
Dimensional Accuracy	$\pm 0.5$ mm	Suitable for educational use
Wi-Fi Connection Time	1–3 seconds	Fast and reliable
Communication Latency	10–20 ms	Near real-time response
Work Area	250 × 250 × 50 mm	Compact, lab-suitable
Wi-Fi Range (indoor)	20–25 meters	No external network needed
Shapes Supported	Square, Rect, Triangle, Circle	Extensible via app
Materials Machined	Foam, MDF, Acrylic, PCB	Soft materials only
Motor Step Resolution	1.8° (1/32 micro-step)	Smooth motion
Spindle Speed	Up to 7000 RPM	Adequate for engraving

#### 6.3.1 Prototype Images

The following images show the fabricated mini-CNC machine prototype from multiple angles, highlighting the mechanical frame, lead screw axes, spindle assembly, control electronics, and the ESP32 and Arduino modules used for wireless operation.

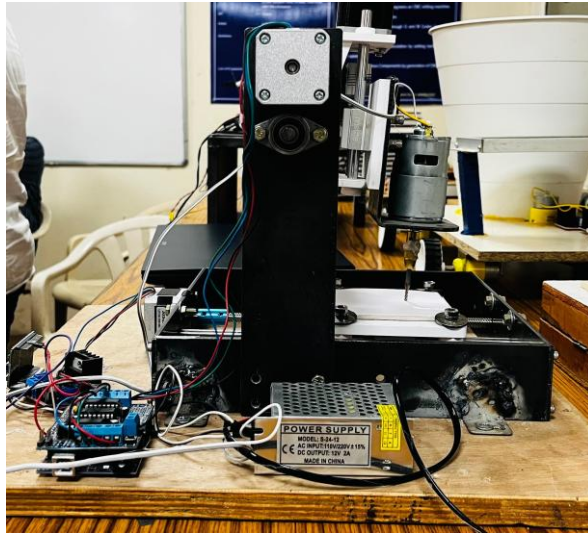


Fig. 1: Smart CNC Machine – Front View with Control Electronics

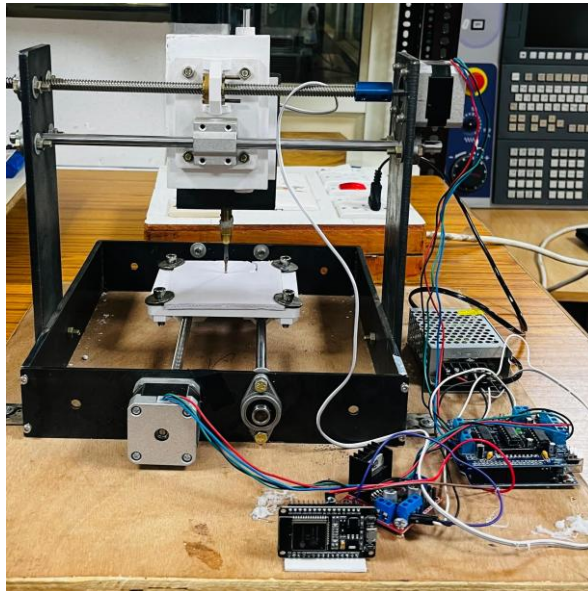


Fig. 2: CNC Machine with ESP32 Module and Arduino Control Board

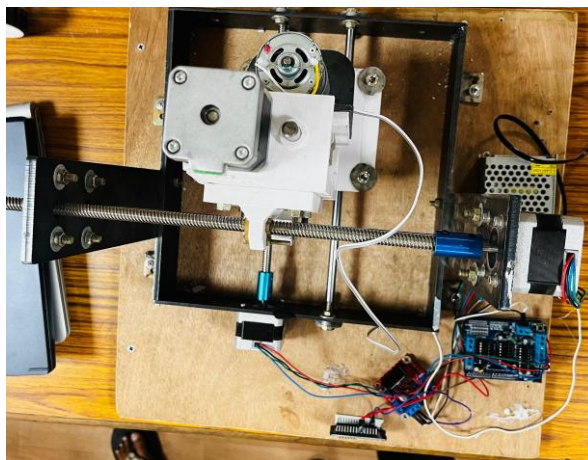


Fig. 3: Top View – Lead Screw Axes and Spindle Assembly

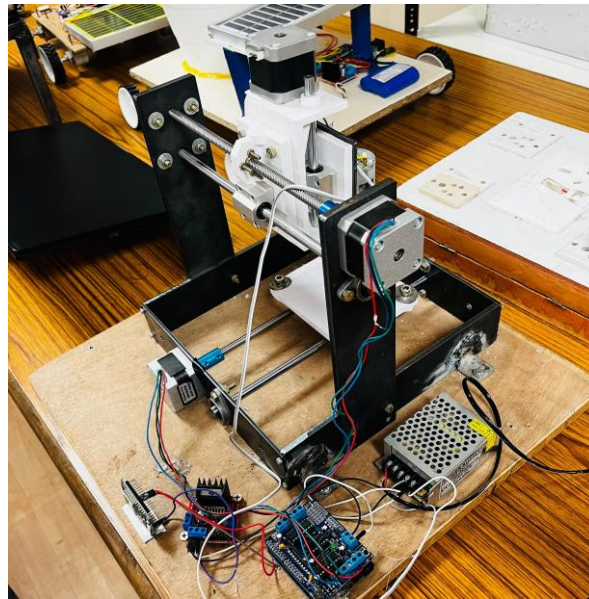


Fig. 4: Three-Quarter View of Complete CNC Prototype

### 6.3.2 Mobile Application Interface

The Android mobile application (CNC Controller) provides a clean dark-themed interface for wireless machine control. The connection status bar at the top tracks the pipeline: Cloud → ESP32 → Arduino. Users select a shape geometry (triangle, square, circle, or pentagon), adjust size in mm and rotation angle using sliders, and view a live preview of the toolpath. Commands are synchronized to the ESP32 in real time, with the action button indicating current status — "Command Synchronized to ESP32" (blue) during active transmission or "Cancel Job" (red) to abort the current operation.

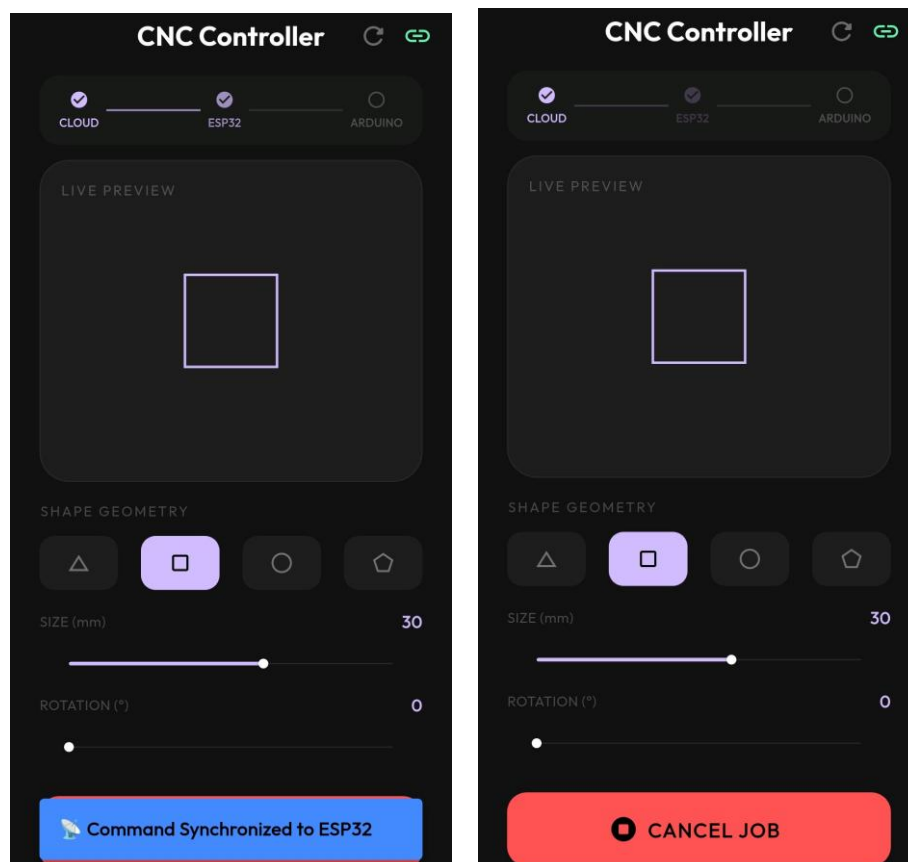


Fig. 5 & 6: CNC Controller App – Live Shape Preview with Command Synchronized (left) and Cancel Job State (right)

### 6.4 Limitations

- Open-frame design introduces minor vibration, reducing rigidity at higher operating speeds.
- Open-loop stepper control without encoder feedback limits real-time error correction capability.
- Machining is restricted to soft materials; the system is not suitable for metal cutting.
- The mobile application currently supports only predefined geometric shapes without custom toolpath import.
- Performance is sensitive to proper axis alignment and GRBL calibration.

## 7. CONCLUSION

This paper presented the design, development, and experimental validation of a smart mini-CNC machine integrating a Wi-Fi-based mobile control application with AI-assisted G-code generation. The system successfully demonstrated automated machining operations with dimensional accuracy within  $\pm 0.5$  mm, stable wireless communication latency of 10–20 milliseconds, and consistent repeatability across multiple test cycles. The mobile application developed in Kotlin significantly simplified CNC operation by eliminating the need for manual G-code programming, making the system accessible to non-expert users including students and hobbyists.

The integration of Arduino with GRBL firmware, ESP32 Wi-Fi communication, and lead screw-based motion control provides an effective and affordable platform for educational CNC demonstration and small-scale prototyping. Although the current system has limitations regarding mechanical rigidity, feedback control, and material range, these are acceptable trade-offs for a low-cost design and can be addressed in future iterations. Future enhancements may include voice-based command input, closed-loop feedback using encoders, cloud-based G-code storage and sharing, support for custom SVG or DXF toolpath import, and extended material capability through improved spindle and frame design. Overall, the project demonstrates the feasibility and effectiveness of combining mechanical design, embedded electronics, wireless communication, and intelligent software into an integrated smart CNC solution aligned with Industry 4.0 objectives.

## REFERENCES

- [1] V. K. Pabolu and K. Srinivas, "Design and Implementation of a Three-Axis CNC Machine," *IJEST*, vol. 2, no. 10, pp. 1–8, 2010.
  - [2] B. Jayachandraiah et al., "Fabrication of Low Cost 3-Axis CNC Router," *IJERA*, vol. 4, no. 8, pp. 31–34, 2014.
  - [3] S. Pandian and S. Rajkumar, "Low Cost CNC Machine Using Arduino and Open Source Software," *IJAREEIE*, vol. 3, no. 6, 2014.
  - [4] T. Singh and N. Rao, "Automation of CNC Machines Using Embedded Controllers," *IJRSET*, vol. 6, no. 5, 2017.
  - [5] M. S. Kumar and R. Sharma, "Low-Cost CNC Machine Development for Academic Institutions," *IJERA*, vol. 7, no. 7, 2017.
  - [6] A. Patel and R. Singh, "Design and Development of a Low-Cost CNC Router," *IJERT*, vol. 7, no. 5, 2018.
  - [7] D. K. Nair and B. Thomas, "Microcontroller-Based Multi-Axis Motion Control Using GRBL Firmware," *IJMETMR*, vol. 6, no. 9, 2019.
  - [8] S. Kumar and V. Rao, "Integration of Stepper Motor Control and Embedded Software in CNC Systems," *IJRET*, vol. 8, no. 3, 2019.
  - [9] P. Karthik and A. Ghosh, "Wireless Data Transmission for Machine Control Using HC-05 Modules," *IJRASET*, vol. 7, no. 6, 2019.
  - [10] K. Dinesh Babu and M. Siva Nagendra, "Wireless CNC Operation Using Arduino and Bluetooth," *IJARME*, vol. 6, no. 2, 2020.
  - [11] G. R. Patel and R. Anand, "Performance Evaluation of Mini CNC Systems for Educational Training," *IJAMT*, vol. 108, 2020.
  - [12] A. Prasad, T. Gupta, and H. Mehta, "Design Optimization of a Three-Axis CNC Plotter for Educational Use," *IJSER*, vol. 11, no. 9, 2020.
  - [13] R. M. Reddy and C. Naresh, "Development of a Mini CNC Milling Machine Using Arduino Controller," *IJRSE*, vol. 5, no. 6, 2021.
  - [14] P. Venkatesh and A. K. Rao, "Implementation of Internet-Enabled CNC Systems for Smart Manufacturing," *IJERT*, vol. 11, no. 1, 2022.
  - [15] S. Vishnu and P. Manoj, "Integration of Mobile Applications with CNC Control Systems," *IJETR*, vol. 11, no. 6, 2023.
  - [16] S. Kumar and A. Vardhan, "Development of a Hybrid Communication Interface for Arduino-Based CNC Machines," *IRJET*, vol. 11, 2024.
  - [17] S. Sarma and R. Lakshman, "Recent Advances in Compact CNC Design for Additive and Subtractive Manufacturing," *IJETAE*, 2025.
- Mahatma Gandhi Institute of Technology (Autonomous) | Gandipet, Hyderabad – 500075 | www.mgit.ac.in*