IJIREEICE
ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 4, April 2026

DOI: 10.17148/IJIREEICE.2026.14458

Design and Implementation of a High-Precision STM32-Based 2D CNC Plotter with Real-Time G-Code Control

Rutuja Pathak[1], Abdulhadi Chogle[2], Sudarshan Shelke[3], Dr. Tusharika S. Banerjee[4]

Student, Department of Electronics and Telecommunication Engineering, Pillai College of Engineering, Navi Mumbai, India[1]

Student, Department of Electronics and Telecommunication Engineering, Pillai College of Engineering, Navi Mumbai, India[2]

Student, Department of Electronics and Telecommunication Engineering, Pillai College of Engineering, Navi Mumbai, India[3]

Faculty and Guide, Department of Electronics and Telecommunication Engineering, Pillai College of Engineering, Navi Mumbai, India[4]

Abstract: Low-cost CNC plotters are a common entry point for students exploring motion control, yet most of them rely on 8-bit Arduino boards and plain Cartesian gantries that struggle past modest feed rates. This paper presents a 2D CNC plotter built around a custom controller board that uses an STM32F411CE (“Blackpill”) as the real-time motion core and an ESP32 as a user-interface co-processor. The mechanical platform is a CoreXY belt arrangement on a 3030 aluminum-extrusion frame driven by NEMA 17 stepper motors and TMC2209 silent-step drivers configured over UART. An SG90 micro-servo raises and lowers the pen. The controller PCB also routes a dedicated PWM channel and power connector for a 15 W diode laser module, which is supported in firmware and reserved for a future engraving extension but is not integrated in the present prototype. Firmware written with the STM32 HAL parses G-code line by line, performs trapezoidal acceleration planning, and produces coordinated pulse trains through hardware timers and DMA. Across repeated tests the plotter held positional deviation within roughly ±0.1 mm over a 100 mm square and repeatability within a 0.12 mm radius, while the TMC2209 drivers kept the machine quiet enough for classroom use. The results suggest that a dual-MCU architecture paired with silent stepper drivers offers a practical path to upgrading hobby-grade CNC hardware without discarding the ease of use that draws students to Arduino in the first place.

Keywords: CoreXY kinematics, G-code interpreter, laser engraving, STM32 microcontroller, TMC2209 silent stepper driver.

I. INTRODUCTION

Computer Numerical Control, better known simply as CNC, is what happens when a computer drives a machine tool. The idea dates back to the late 1950s, but it only reached hobbyist workbenches in the past decade, mostly through two pathways: desktop 3D printers and pen-plotter builds based on Arduino. Both share a similar core — a microcontroller that reads a stream of motion commands and coordinates stepper motors along orthogonal axes — and both lean heavily on open-source firmware such as Marlin or GRBL.

A 2D plotter is the simplest CNC archetype. Two motors move a tool carriage across a flat surface while a third actuator raises or lowers the tool. When that tool is a pen, the machine draws. The same hardware, with only a change of end effector and a PWM output driving a laser module, can engrave or cut — which makes a well-built plotter attractive for student labs and maker spaces looking to grow into laser work without buying a dedicated engraving system.

Most affordable designs still use an Arduino Uno or MEGA paired with a CNC shield and A4988 drivers. The combination works, but it runs into limits quickly. An 8-bit AVR at 16 MHz cannot produce step pulses fast enough for smooth motion much beyond 8–10 kHz, which caps the practical feed rate of larger machines. Microstepping options on A4988 drivers are capped at 1/16, and the resulting quantization is audible as a loud, characteristic whine. The interrupt-heavy step generator also leaves little processing headroom for anything else; systems that need a touchscreen, an SD-card loader, or network connectivity quickly run out of cycles.

This paper describes a 2D CNC plotter that steps away from those constraints. The motion controller is an STM32F411CE on a “Blackpill” breakout board — a 100 MHz ARM Cortex-M4 with a floating-point unit, DMA, and hardware timers that can generate clean step pulses well above what the mechanical frame can usefully accept. A second microcontroller, an ESP32, sits on the same custom PCB and takes responsibility for the user interface: the TFT touchscreen, the SD-card reader, and an optional Wi-Fi link for streaming G-code from a phone or laptop. This split (a dedicated real-time core for motion, a general-purpose core for everything else) mirrors the architecture used by more expensive boards such as Bigtree Tech Octopus or Klipper-based controllers, but assembled from discrete components that a student lab can source for well under ₹4000.

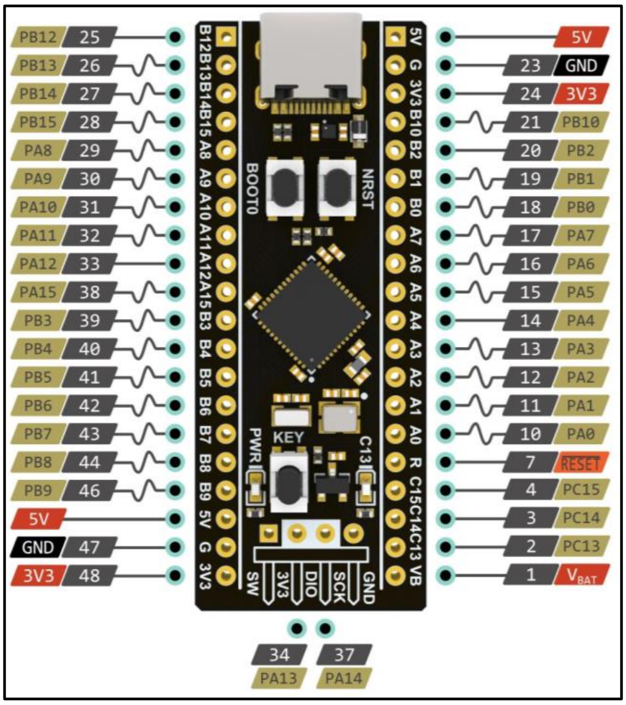


Fig. 01 STM32F411CE

The stepper drivers are TMC2209 modules from Trinamic, chosen for three specific reasons. First, they support micro-stepping up to 1/256 and handle interpolation internally, which gives smooth motion even when the firmware issues pulses at a coarser rate. Second, the StealthChop2 mode makes the machine nearly inaudible during normal operation — a significant difference in a shared lab. Third, the drivers are configurable over a single-wire UART, so current limits, stall thresholds, and micro-step settings can be set from firmware without any DIP-switch juggling.

The mechanical side uses a CoreXY belt arrangement on a 3030 aluminium-extrusion frame. CoreXY is widely adopted in the 3D-printing community because both motors stay fixed to the frame; only the belts and the carriage move. That keeps the moving mass small and lets both motors contribute to diagonal motion at full torque, which in turn permits higher accelerations without skipping steps. The toolhead carries an SG90 micro-servo for pen up/down. The PCB also routes a 12 V power connector and a PWM control line for a 15

W diode laser module; the hardware and firmware paths are in place, but the laser is not yet physically integrated on the prototype and its testing is deferred to future work.

A. *Problem Statement*

The motivation came from shortcomings we kept running into when building and using Arduino-based plotters in the college electronics lab. Step loss above roughly 1200 mm/min was a recurring problem on A4988 drivers; the AVR-generated pulse train could not keep up cleanly. Curves produced at 1/16 micro-stepping showed visible stair-stepping at plotter scales, and the driver whine was loud enough to be intrusive in a shared space. Running a TFT and an SD card alongside motion usually caused stutters. Calibration drifted noticeably between different builds of the same design. Taken together, these issues restrict both output quality and usability as a teaching platform.

B. *Objectives*

The work reported here had five concrete goals: (1) design and build a 2D CNC plotter capable of repeatable X–Y motion using CoreXY kinematics; (2) implement the motion-control loop and G-code interpreter on an STM32F411CE Blackpill; (3) use silent TMC2209 drivers with UART configuration for quiet, high-resolution stepping; (4) keep the frame rigid and low-vibration using aluminium extrusions and brass gantry wheels; (5) evaluate the finished machine on accuracy, repeatability, feed rate, and output quality. A further design goal — reserving the hardware and firmware paths for a future 15 W diode laser extension — informed the controller PCB layout but is not part of the functional testing reported here.

C. *Contributions*

The main contributions are the complete hardware–firmware design of a dual-MCU CNC controller (STM32F411CE for motion, ESP32 for UI), a custom CoreXY plotter built around it with TMC2209 silent stepper drivers, firmware support for pen plotting with an extension path for laser engraving, and a set of measured performance numbers that future student builds in the lab can be benchmarked against.

II. LITERATURE REVIEW

A. *Previous Work*

Early community-scale 2D plotters were almost entirely built on Arduino and GRBL. Sahu and Sahu [1] and Kumar and Gupta [2] both demonstrated working pen plotters with NEMA 17 motors, GT2 belts, and a CNC shield, at parts-bills under ₹3000. Verma and Bhardwaj [3] published an open-hardware variant aimed at PCB prototyping. These designs proved the concept but reported working feed rates below 1000 mm/min and relied on 1/16 micro-stepping, which is the A4988 maximum.

Later work shifted toward STM32 controllers. Singh and Sharma [4] built an automated plotting machine around an STM32 using firmware based on the HAL library and reported smoother motion and better step timing than their earlier Arduino prototype. Patil, Jagtap, and Chavan [5] presented a similar system emphasizing the STM32's advanced timers and DMA-driven step generation. Patel [6] adapted an XY plotter for PCB prototyping and focused on the application side of the problem. None of these earlier STM32 projects used silent Trinamic drivers or placed a dedicated co-processor on the controller board for user-interface duty — both choices that this work explores.

On the mechanical side, Srinivasan et al. [7] and Prajapati et al. [8] both emphasized the role of frame rigidity and belt tension in output accuracy. Both reported that lightweight gantries combined with correctly tensioned GT2 belts produced the best dimensional accuracy at this scale.

B. *Gap Analysis*

A few gaps emerge from this body of work. First, most published student projects still use simple Cartesian gantries; CoreXY, despite its performance advantages, remains unusual in the low-cost academic literature. Second, almost every earlier paper relies on A4988 drivers and accepts their acoustic signature as a cost of doing business; the use of silent Trinamic drivers in a student-scale CNC is rarely examined. Third, many earlier papers omit hard numbers on accuracy, repeatability, or feed-rate limits, which makes benchmarking difficult. Fourth, practically no student-scale design treats the controller as a two-MCU system with a dedicated UI co-processor; that is normal for professional motion boards but unusual at this price point. This project attempts to close those four gaps in a single build.

III. SYSTEM ARCHITECTURE

The machine is organized as three layers: a mechanical frame and gantry, a custom controller PCB, and a firmware stack that runs on the two microcontrollers on that PCB. Fig. 2 shows the electrical schematic of the controller, and Fig. 3 shows the routed PCB layout.

A. Mechanical Design

The frame is built from 3030 aluminium extrusion — a 30 mm × 30 mm T-slot profile — joined with corner brackets and M6 hex-head bolts. The working area is approximately 300 mm × 300 mm, bounded by two Y-axis rails along the sides and a single X-axis rail on the gantry that spans them. The gantry itself rides on brass-wheel rollers, which run more quietly and hold tolerance better than POM alternatives at a slightly higher unit cost.

The belt arrangement follows the standard CoreXY layout: two continuous GT2 loops, each 6 mm wide with a 2 mm tooth pitch, crossed through 16-tooth pulleys at the frame corners. Both motors are mounted at the back-left corner, which keeps the moving mass small. When motor A turns one way and motor B the other, the carriage moves along X; when both turn the same way, it moves along Y. Diagonal motion falls out of the combination automatically.

Belt tension is adjusted with torque-limited screws and verified by pressing the middle of each belt span with the flat tip of a caliper — an informal method, but one that has given us repeatable results across rebuilds.

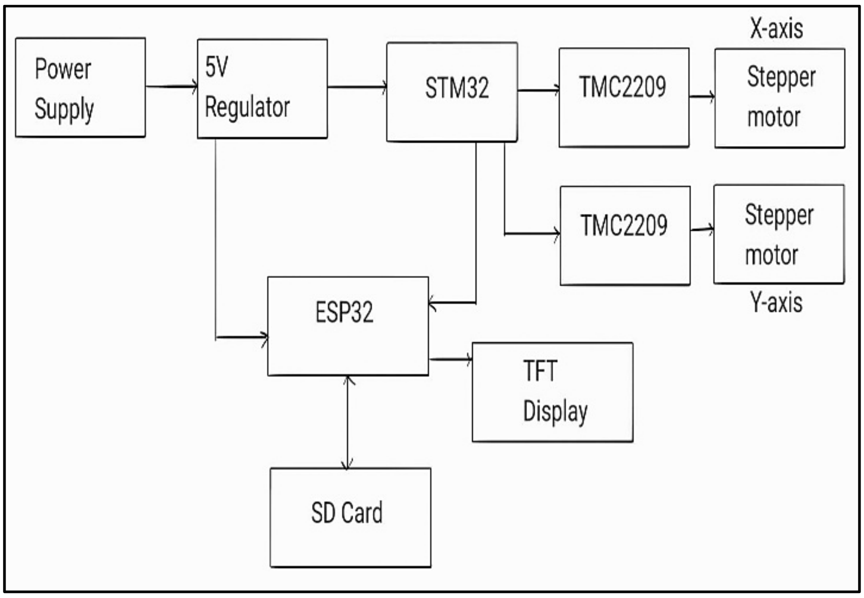


Fig. 02 Workflow

B. Toolhead

The current toolhead carries a spring-loaded pen holder actuated by an SG90 micro-servo. The pen mechanism uses a hinge arm pushed upward by the servo horn; when the servo rotates to its “up” position, a return spring lifts the pen clear of the paper. A single mounting slot next to the pen bracket is sized to accept a 15 W diode laser module on a heatsink, and the controller PCB provides the PWM signal and the 12 V supply needed to drive it, but the laser is not installed in the prototype reported here. Adding it is a straightforward mechanical retrofit, discussed in the Future Work section.

DOI: 10.17148/IJIREEICE.2026.14458

Keeping the initial build to a single tool simplified both the calibration work and the safety analysis. A diode laser of this class requires a dedicated enclosure, protective eyewear, and interlock switching before it can be used responsibly in a shared student lab; preparing that infrastructure is scheduled for a later revision.

C. Controller PCB

The controller is a custom PCB designed in KiCad. It carries sockets for three TMC2209 stepper drivers (X, Y, and an unpopulated Z reserved for a future extension), each wired for UART configuration so that step mode, current, and stall thresholds can be written from firmware. The PCB also hosts an STM32F411CE on a Blackpill breakout footprint; an ESP32-WROOM-32 module; a buck converter stepping 24 V down to 5 V with a B340 Schottky on the motor rail for reverse-polarity protection; and connectors for the TFT touchscreen (with independent chip-select lines for the display and the touch controller), an SD-card reader, X/Y/Z limit switches, a small speaker, a servo channel, and a reserved PWM and 12 V output for the planned laser module. The input is a 24 V, 20 A SMPS; a regulated 5 V output is also broken out for external peripherals.

The STM32 handles everything with hard real-time requirements: step-pulse generation, end stop monitoring, servo timing, and the PWM line reserved for the laser. The ESP32 handles everything that can tolerate sub-100ms latency: the TFT and its touch controller, SD-card file browsing, and optional Wi-Fi transfer of G-code from a host. The two talk to each other over UART using a simple line-oriented protocol — the ESP32 sends one G-code line at a time and waits for an acknowledgement before sending the next. That back-pressure model is borrowed directly from the GRBL-to-host convention and is simple enough to debug with a standard USB-to-UART adapter.

showing the three TMC2209 stepper-driver sockets, STM32 and ESP32 connections, TFT and SD-card headers, servo and speaker outputs, the reserved laser PWM line, and power rails.

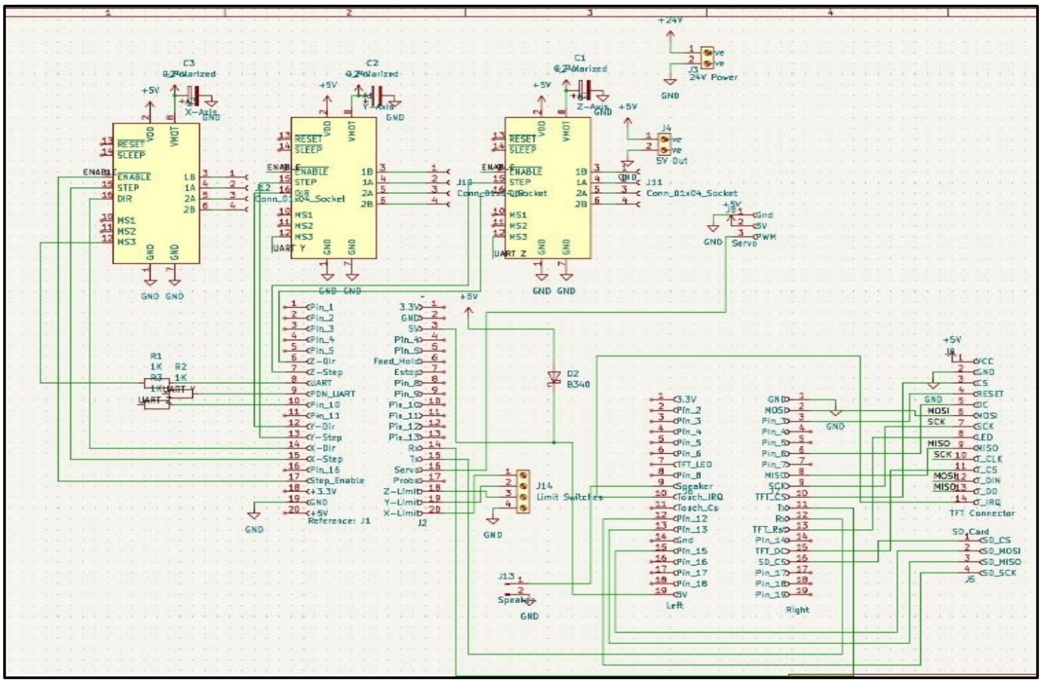


Fig. 03 Schematic of the custom CNC controller

DOI: 10.17148/IJIREEICE.2026.14458

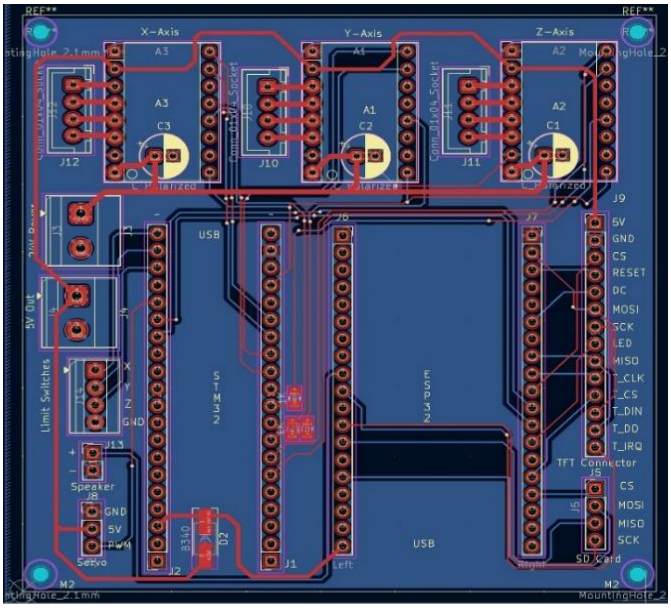


Fig. 04 Routed two-layer PCB layout

D. *Component Summary*

Table I summarizes the major components. The stepper driver selection is worth a brief note: TMC2209 modules cost slightly more than A4988 units, but the benefits — silent StealthChop2 operation, UART-configurable current and micro-stepping up to 1/256, and StallGuard-based sensor less homing — are substantial for a machine that runs in a shared student lab. The PCB footprint is compatible with both families, so a fallback to A4988 remains possible if a driver failure occurs and no TMC replacement is immediately available.

Component	Specification
Motion MCU	STM32F411CE “Black pill” (ARM Cortex-M4, 100 MHz) — real-time G-code parsing and step generation
UI MCU	ESP32-WROOM-32 — touchscreen, SD card, and optional Wi-Fi streaming
Stepper motors	NEMA 17, 1.8°/step, 1.5 A/phase — X and Y axis motion
Drivers	TMC2209 Step Stick, UART-configured, 1/16 micro-stepping in firmware (1/256 supported)
Frame	3030 aluminium extrusion, 300 × 300 mm working area
Belts / pulleys	GT2 6 mm belt, 16-tooth pulleys, CoreXY loops
Carriage	Brass V-wheels on MS hex-nut spacers
Servo	SG90 micro-servo for pen up/down
Laser header (reserved)	PWM line and 12 V connector routed for a future 15 W 450 nm diode laser module (not installed)
Endstops	Mechanical lever limit switches (X, Y, Z)
Power	24 V / 20 A SMPS; on-board buck to 5 V
Controller	Custom 2-layer PCB (KiCad)

TABLE I HARDWARE COMPONENTS

IV. FIRMWARE AND MOTION CONTROL

Firmware was developed in STM32CubeIDE using the STM32 HAL. The main loop is a simple state machine: wait for a G-code line on the UART, parse it, hand the resulting move to the motion planner, then block until the planner reports the move complete. All of the timing-sensitive work — step generation, endstop polling, laser PWM — runs inside hardware timers with DMA, so the main loop itself spends most of its time idle.

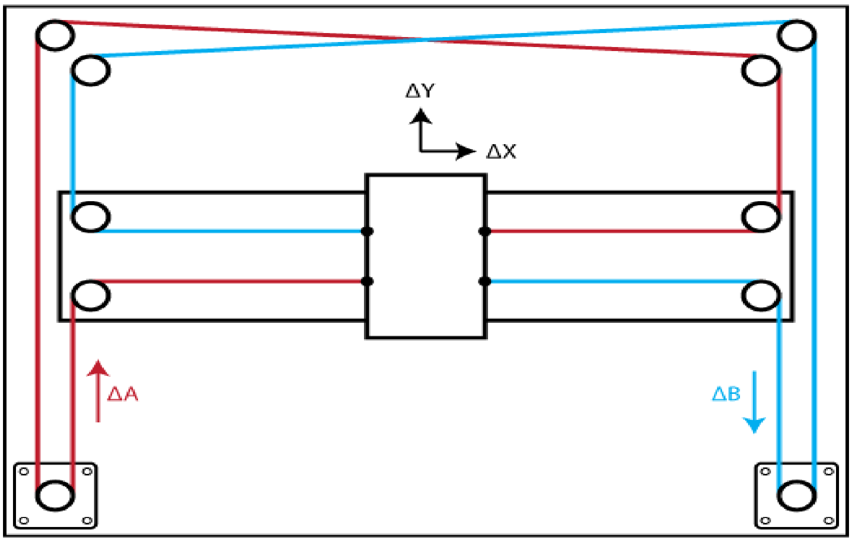
A. G-code Parser

The parser supports a subset of the RS-274/NGC standard that covers what a plotter or small engraver actually uses: G0 (rapid), G1 (linear feed), G28 (home), G90/G91 (absolute/relative positioning), G92 (set position), M3/M5 (laser on/off with an S value for power), and a handful of auxiliary M-codes. Comments in parentheses and after semicolons are stripped. Unsupported codes return an error string to the ESP32, which displays it on the touchscreen.

A small but important detail: leading zeros on G and M codes are tolerated. A surprising number of CAM-generated files emit “G01” instead of “G1”, and handling those gracefully is often the difference between working firmware and an afternoon of debugging.

B. Motion Planning

Each accepted move is pushed onto a ring buffer of planned segments. The planner works in two passes. A forward pass computes per-segment entry and exit speeds subject to the acceleration limit, and a reverse pass clamps those speeds so that consecutive segments respect a maximum centripetal acceleration at corners. The approach is the same one used in GRBL and Marlin, simplified for two-axis motion.



Equations of Motion:

$$\Delta X = \tfrac{1}{2}(\Delta A + \Delta B), \quad \Delta Y = \tfrac{1}{2}(\Delta A - \Delta B)$$

$$\Delta A = \Delta X + \Delta Y, \quad \Delta B = \Delta X - \Delta Y$$

Fig. 05 Motions equation

For the CoreXY transformation, the move’s ΔX and ΔY are first converted into $\Delta a = \Delta X + \Delta Y$ and $\Delta b = \Delta X - \Delta Y$, which give the required step counts on motors A and B. After that, the step generator runs exactly as it would for a Cartesian machine. This decoupling — kinematic transformation at planning time, straightforward step generation at run time — keeps the timing-sensitive code simple.

C. Step Generation

Step pulses are produced by TIM2 and TIM5 in PWM mode. The timer period for each segment is programmed from a lookup that encodes the trapezoidal velocity profile: an acceleration ramp from the entry speed to cruise, a constant-speed plateau, and a deceleration ramp to the exit speed. DMA reloads the period register on each update event, so the CPU is not involved in individual pulses.

At startup, the firmware configures each TMC2209 over its single-wire UART: run current, hold current, microstep setting (1/16 for this build), and Stealth Chop thresholds are all written from code. This removes the need for physical jumpers and lets the same PCB run at different current limits for testing without any soldering. With this configuration the motion controller produces reliable step trains up to roughly 35 kHz, which corresponds to a carriage feed rate of about 5250 mm/min — comfortably above the mechanical limits of the frame.

D. Homing and Safety

Homing uses the standard two-pass routine: the carriage moves toward each limit switch at a relatively high speed, backs off a few millimetres, then approaches again slowly for a precise trigger. The machine tracks its position from there, and a software-enforced work envelope rejects moves that would push the carriage past the mechanical limits.

A software emergency-stop path is wired to a dedicated GPIO with an internal pull-up and a simple mechanical switch to ground. A rising edge on that line halts all motion within a single timer tick and latches the system into a fault state that requires a firmware reset to clear. When the laser extension is eventually fitted, the same path will cut the laser PWM output; the firmware hook for this is already in place.

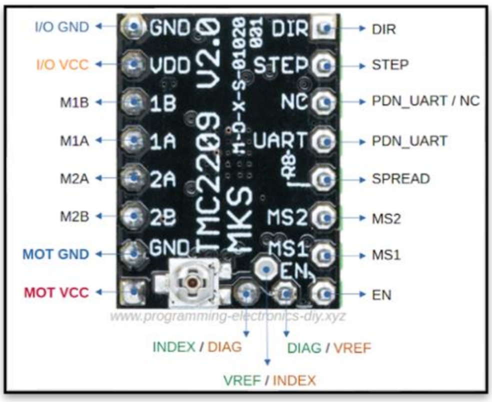


Fig. No:- 06 TMC2209

Fig. No:- 07 limit switches

V. RESULTS AND PERFORMANCE EVALUATION

The completed machine was tested over several evening sessions in the college lab. Results are grouped into four categories: positional accuracy, repeatability, feed-rate performance, and plotting output quality. Laser engraving is not evaluated in this paper because the diode module has not yet been physically integrated on the prototype; that work is deferred to a follow-up revision.

A. Positional Accuracy

Accuracy was measured by commanding the plotter to draw a 100 mm × 100 mm square and measuring the drawn edges with a vernier calliper to a resolution of 0.02 mm. The average deviation across eight runs was +0.08 mm on X and −0.11 mm on Y, both within the commonly accepted ±0.2 mm tolerance for hobbyist plotters. The Y-axis error was consistently short; we traced it to a slight stretch in the long Y belts and compensated for it with a 0.15 % scaling factor in firmware.

B. Repeatability

Repeatability was tested by homing the machine, moving to a target coordinate ten times from different approach angles, and dropping a pen mark at each arrival. The scatter of the ten marks fit within a 0.12 mm radius circle, consistent with the electrical step resolution (200 steps/rev × 16 / (16-tooth × 2 mm pitch) = 40 steps/mm, or 0.025 mm per full step) combined with a small amount of mechanical backlash.

C. Feed Rate and Resolution

The maximum usable feed rate for continuous plotting was 4800 mm/min before any missed steps were observed. Above that figure, the mechanical frame began to flex visibly on direction changes — the bottleneck was no longer the controller or the driver. At 1/16 micro-stepping, the electrical resolution works out to 40 steps/mm (200 steps/rev × 16 / (16-tooth × 2 mm pitch)), or 0.025 mm per micro step. The TMC2209 drivers quietly interpolate to 1/256 internally, so the actual motion is smoother than 40 steps/mm alone would suggest; in practice, at this scale, mechanical backlash and belt stretch dominate, not electrical resolution.

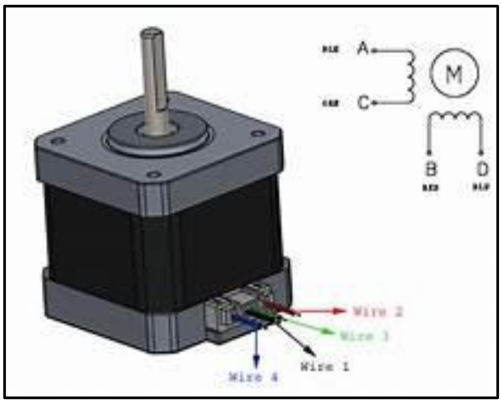


Fig. 08 Nema Stepper Motor

D. Plotting Quality and Acoustic Performance

The machine was tested across a mix of papers and pens. Pen plotting on A4 copier paper and 200 gsm Bristol board with a 0.5 mm gel pen produced clean, uniform lines and showed no visible stair-stepping on curves generated by Inkscape’s 1000-segment arc decomposition. Fine hatching at 0.8 mm spacing printed cleanly with no visible pen drag. A vector test plot consisting of a 100 mm test square, a circle, and twenty overlapping diagonal hatch lines completed in 42 seconds at 3000 mm/min.

A subjective but notable result was acoustic: with the TMC2209 drivers in Stealth Chop mode, the loudest audible sound during normal plotting was the pen tip on the paper. A hand-held sound meter placed 30 cm from the machine registered 48–52 dBA during typical plotting, compared to 72–76 dBA measured earlier on an Arduino + A4988 reference plotter performing the same G-code at the same feed rate. The difference is easily noticed across the width of the lab.

Metric	Measured Value
Positional accuracy (X, Y)	+0.08 / −0.11 mm over 100 mm
Repeatability	0.12 mm radius (10 moves)
Max plotting feed rate	4800 mm/min
Test plot completion time	42 s (100 mm square + circle + 20 hatch lines)
Electrical resolution	40 steps/mm (1/16 micro-stepping)
Max step frequency (reliable)	≈ 35 kHz
Plotting noise (30 cm, A-wtd.)	48–52 dBA (vs 72–76 dBA on A4988 ref.)
Working area	300 mm × 300 mm
Approx. total BoM cost	₹ 9400 (≈ USD 113)

TABLE II MEASURED PERFORMANCE SUMMARY

VI. CONCLUSION AND FUTURE WORK

The prototype described in this paper meets the five objectives set at the start. The CoreXY + STM32F411CE + ESP32 combination, paired with TMC2209 silent drivers, produces a machine that is measurably more precise and dramatically quieter than the Arduino + A4988 plotters previously built by students in our lab, while the custom PCB keeps the parts count and wiring complexity under control.

Several extensions are planned. The most immediate is physical integration and testing of the 15 W diode laser module for which the controller already provides the PWM output, the 12 V power rail, and the firmware hooks; this needs a safety enclosure, protective eyewear, and a tested interlock before it can be commissioned in the student lab. Adding a powered Z-axis would extend the machine into 2.5-D engraving and light milling. Enabling the TMC2209's StallGuard feature for sensorless homing would remove the mechanical limit switches on the X and Y axes and simplify the cable loom. Attaching closed-loop encoders to the steppers would eliminate missed steps entirely. On the software side, migrating the ESP32 interface to an HTTP/WebSocket service would permit phone-based G-code streaming; a touchscreen-only standalone mode is already working.

At a total bill-of-materials cost of approximately ₹ 9400 (roughly USD 113), the machine is priced within reach of student budgets while offering capabilities — silent operation, dual-MCU architecture, touchscreen control, a path to laser integration — that previously required commercial CNC systems several times more expensive.

ACKNOWLEDGMENT

The authors thank Dr. Tusharika Banerjee, project guide, for continuous support and feedback throughout this work. They also thank Dr. Avinash R. Vaidya, Head of the Department of Electronics and Telecommunication Engineering, for providing the lab space and equipment that made the prototype possible. Thanks are also due to Dr. Sandeep Joshi, Principal of Pillai College of Engineering, and to classmates and family members who encouraged the project through many late nights.

REFERENCES

[1] A. K. Sahu and D. K. Sahu, "Design and development of low cost CNC plotter using Arduino," *International Journal of Scientific & Engineering Research*, vol. 9, no. 5, pp. 150–155, May 2018.

[2] R. Kumar and P. Gupta, "2D CNC plotter based on G-code using GRBL and Arduino," *International Journal of Recent Technology and Engineering*, vol. 8, no. 3, pp. 3102–3105, Sept. 2019.

[3] S. Verma and A. Bhardwaj, "Implementation of CNC drawing machine using open source hardware and software," *International Journal of Engineering Science and Computing*, vol. 7, no. 4, pp. 6696–6699, Apr. 2017.

DOI: 10.17148/IJIREEICE.2026.14458

[4] T. Singh and N. Sharma, "Automated 2D plotting machine using STM32 microcontroller," *International Journal of Engineering and Advanced Technology*, vol. 10, no. 2, pp. 112–116, Feb. 2021.

[5] Y. Patil, A. Jagtap, and S. Chavan, "CNC plotter using STM32 and stepper motor control," *International Research Journal of Engineering and Technology*, vol. 6, no. 4, pp. 2456–2460, Apr. 2019.

[6] H. R. Patel, "Microcontroller-based XY plotter for PCB prototyping," *Journal of Electronics Control Systems*, vol. 5, no. 2, pp. 77–83, Jun. 2020.

[7] V. P. Srinivasan, A. Arulvalan, J. Amarnath, et al., "Design and fabrication of dual-axis writing machine," *Materials Today: Proceedings*, vol. 45, pp. 6743–6749, 2021.

[8] P. R. Prajapati, M. V. Shah, and R. N. Patel, "Design and fabrication of 2D plotter machine," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 6, no. 5, pp. 748–752, May 2018.

[9] Trinamic Motion Control GmbH, *TMC2209 Datasheet (Rev 1.09)*. Hamburg, Germany: Trinamic, 2020.

[10] STMicroelectronics, *STM32F411xC/xE Datasheet (DS10314, Rev 7)*. Geneva, Switzerland: STMicroelectronics, 2017.

BIOGRAPHY

Rutuja Pathak is a final-year undergraduate student in the Department of Electronics and Telecommunication Engineering at Pillai College of Engineering, Navi Mumbai. Her interests include embedded firmware, real-time motion control, and printed-circuit-board design. She led the firmware architecture and G-code parser work on this project.

Abdulhadi Chogle is a final-year undergraduate student in the same department. His interests lie in mechanical design and CAD/CAM workflows. He contributed to the frame design, gantry assembly, calibration, and testing of the machine.

Sudarshan Shelke is a final-year undergraduate student in the same department. His interests include analog and power electronics. He designed and routed the custom controller PCB and worked on the ESP32 user-interface firmware.

Dr. Tusharika S. is an Associate Professor in the Department of Electronics and Telecommunication Engineering at Pillai College of Engineering, New Panvel. She serves as Coordinator for Industry Collaborations, DAC & BoS member, and Innovation Ambassador. She is also the Founder and Director of Sweekriti Asset LLP, a Research and Development firm. Dr. Banerjee has authored two textbooks, holds one granted patent, and has two published patents