AN-NAJAH NATIONAL UNIVERSITY

Computer Engineering Department

Hardware Graduation Project

CNC Foam Cutting Machine

ScuptoFoam

Students
Mohammed Jaddou and Amjad Mousa

Under the supervision of
Dr.Hanal Abu-Zant

A report submitted in partial fulfillment of the requirements for
bachelor's degree in computer engineering in the Faculty of Engineering
Information Technology - Hardware Project

June 2025

Acknowledgements

We would like to express our gratitude to everyone who contributed to this initiative. It would not have been possible if it weren't for their assistance and support. Many thanks to Dr. Hanal Abu-Zant for his leadership, tenacious oversight, and assistance in seeing the project through to completion. Our families' support during this trying and stressful period, our coworkers' moral and technical support, and the people who volunteered their time are all greatly appreciated.

Disclaimer

The authors of this research are Mohammed Jaddou and Amjad Mousa from the Faculty of Engineering's Department of Computer Engineering at An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the authors. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Contents

List of Figures

Abstract

Cutting foam for packaging, insulation, and prototyping requires precision and efficiency, but traditional approaches such as manual cutting or bulky industrial CNC machines present significant challenges. Manual cutting often lacks accuracy and consistency, while existing CNC systems are expensive, power-hungry, and typically dependent on a laptop or external computer for operation. This project addresses these limitations by introducing a portable and standalone CNC foam cutting machine that combines affordability, usability, and flexibility for both 2D and 3D cutting applications.

The proposed system integrates electronics, software, and mechanical components into a compact unit. An Arduino Uno controls three stepper motors, each corresponding to the X, Y, and Z axes, while a DC-DC converter regulates current to a heated wire used as the cutting tool. A Raspberry Pi with an attached monitor replaces the traditional PC interface, allowing direct standalone operation. In addition, a Flask-based mobile/web application provides remote control over Wi-Fi, enabling users to jog the machine, upload G-code files, and monitor cutting progress in real time.

Key hardware elements include three NEMA stepper motors, a heated wire with adjustable temperature control, a DC-DC converter, and a custom-built frame. On the software side, a desktop GUI on the Raspberry Pi handles G-code visualization and manual controls, while the Flask server delivers a responsive and mobile-friendly interface. Together, these systems enable precise and automated cutting of foam into 2D shapes and 3D structures without the need for external computers.

Despite challenges such as limited availability of components, fragility of 3D-printed couplers, and ensuring reliable serial communication between the Raspberry Pi and Arduino, the prototype was successfully built and tested. Results demonstrate accurate and efficient cutting, confirming the machine’s potential as an educational and prototyping tool.

The modular and scalable design of this CNC foam cutter opens the door for future enhancements, such as extending compatibility to plastics, Integrate camera/vision system for monitoring, Automated material handling. This project demonstrates how embedded systems and automation can be applied to create cost-effective and standalone manufacturing solutions, bridging the gap between DIY approaches and industrial CNC equipment.

1 Introduction

Foam is one of the most widely used materials in industries such as packaging, insulation, and prototyping because it is lightweight, easy to handle, and cost-effective. Despite its advantages, shaping foam with precision remains a significant challenge. Manual cutting, which is still common in small workshops and educational environments, is inefficient, inconsistent, and unsuitable for complex geometries. On the other hand, industrial CNC machines offer high accuracy and repeatability but are expensive, bulky, and often tied to external computers for operation. These limitations create a gap for a solution that is both **accessible and efficient**, capable of delivering precise cuts without the drawbacks of cost and complexity.

This project introduces a **standalone CNC foam cutting machine** designed to fill that gap. By integrating hardware, electronics, and software into a compact and modular platform, the system provides both **2D plane cutting** and **3D shaping** capabilities. The machine is powered by an **Arduino Uno** that drives three stepper motors—each dedicated to one axis of motion—while a **DC-DC converter** regulates the current supplied to a heated wire, which serves as the cutting element. To remove the dependency on laptops, a **Raspberry Pi with an integrated display** handles direct machine control. For added flexibility, a **Flask-based mobile/web application** enables remote operation, including jogging the machine, uploading G-code files, and monitoring real-time progress.

Beyond its technical implementation, this project emphasizes **affordability, portability, and educational value**. The system demonstrates how a relatively simple combination of open-source software and widely available hardware can replicate functions typically reserved for industrial-grade CNC machines. Its potential applications range from classrooms and makerspaces to small-scale prototyping environments, making advanced foam fabrication more accessible.

2 Constraints and Earlier course work

2.1 Constraints

During the development of this project, several practical constraints emerged that impacted the design and implementation process:

- **Limited Hardware Availability**: Many of the essential components were difficult to source locally. In some cases, alternatives of lower quality had to be used, which affected performance and reliability. Delays in ordering and shipping also slowed down the assembly process.
- **Fragile 3D-Printed Couplers**: To connect the stepper motors to the machine's mechanical frame, we designed and fabricated couplers using 3D printing. However, these couplers often proved too fragile under load, breaking during testing and requiring repeated redesigns and replacements. This not only consumed valuable time but also limited the machine's robustness.
- **Communication Reliability**: Establishing smooth serial communication between the Raspberry Pi and the Arduino Uno proved challenging. Buffering issues, occasional delays, and dropped commands introduced instability in the cutting process, especially during longer G-code executions. Debugging and fine-tuning these communication protocols took considerable effort.

2.2 Early Courses

While all courses in our academic journey contributed to our overall engineering foundation, certain ones had a particularly strong influence on the development of the CNC foam cutting machine. These courses provided a balanced mix of theoretical knowledge and hands-on practical experience, enabling us to design a system that was both innovative and feasible.

Digital Systems and Microcontrollers courses introduced us to embedded programming and hardware control, forming the basis for our understanding of microcontroller systems such as the Arduino Uno. The practical lab sessions, especially with Arduino-based projects, helped us develop the skills to interface stepper motors, sensors, and converters, **Microcontrollers Lab** course refined our ability to integrate software with hardware, teaching us how to handle communication protocols, serial interfaces, and real-time task management—all critical in linking the Raspberry Pi and Arduino Uno, **Electronics and Circuits** A solid grounding in electronics allowed us to design safe and functional connections between the power supply, DC-DC converter, and heated wire. Understanding current regulation, resistance heating, and power distribution was essential for ensuring system safety and reliability, **CPU Lab** course equipped us with debugging techniques, wiring, and assembly skills, all of which were invaluable when dealing with communication errors and fragile hardware components during prototyping, the **Critical Thinking and Scientific Research** course played a vital role in improving our research capabilities and our ability to interpret and apply technical documentation effectively. Finally, the **Technology Project Management** course was instrumental in shaping the business and strategic aspects of the project. It introduced us to market analysis, feasibility studies, cost management, and overall project planning—ensuring that our system could be considered not just as a technical prototype but as a market-ready product.

Together, these courses created the backbone of our technical and managerial knowledge. They enabled us to approach the CNC foam cutter not just as a prototype, but as a **scalable and educational tool** with real potential for practical application.

3 Methodology

The methodology for developing the portable CNC foam cutting machine was structured to achieve a precise, efficient, and user-friendly system for both 2D and 3D foam shaping. The project integrates mechanical design, electronics, and software into a compact, standalone unit that operates without an external computer.

3.1 Standards and Specifications

In designing the control and communication systems, we adopted relevant engineering standards to ensure reliability and interoperability. The system uses standard USB serial communication (based on protocols like UART) for the critical link between the Raspberry Pi and the Arduino Uno. This ensures consistent and predictable command transmission for motor control. For wireless operation, the system complies with the IEEE 802.11 standard for Wi-Fi communication, allowing the Raspberry Pi to host a web server and provide a stable connection for the mobile control interface.

All electrical components were selected to operate within a 12V DC system for the stepper motors and a separately regulated lower voltage for the heated nichrome wire, aligning with standard low-power and workshop safety practices. The G-code parsing and generation adhere to industry-standard (ISO 6983) command sets to ensure compatibility with popular CAD/CAM software.

3.2 Design Overview

The project is divided into three main subsystems:

1. **Motion Control System:** This is comprised of three NEMA stepper motors, each driving a single axis (X, Y, and Z) via threaded rods. An Arduino Uno, paired with a CNC shield and stepper motor drivers, translates high-level commands into precise step and direction signals to coordinate the 3D movement of the cutting head.
2. **Cutting and Power System:** The cutting tool is a heated nichrome wire. Its temperature is regulated by a DC-DC converter module, which adjusts the current flow to the wire based on user input, allowing for clean cuts through different foam densities. A separate power supply provides ample current for both the motors and the heating element.
3. **Software and Control Interface:** A Raspberry Pi serves as the standalone brain of the machine. It runs a desktop GUI for direct control and a Flask-based web server. This dual-interface approach allows users to either operate the machine directly via the attached touchscreen or remotely through any Wi-Fi-connected device. The Raspberry Pi sends high-level G-code commands to the Arduino Uno for execution.

3.3 Hardware Design (Modeling Phase)

Before physical assembly, key mechanical components and the overall frame were designed and tested using 3D modeling software. This phase was crucial for validating part fit, identifying potential collisions, and optimizing the design for stability and ease of assembly. Components such as motor brackets, wire tensioners, and couplers were modeled, 3D printed, and iteratively refined. Below is an image showing the initial hardware design created during this modeling phase:

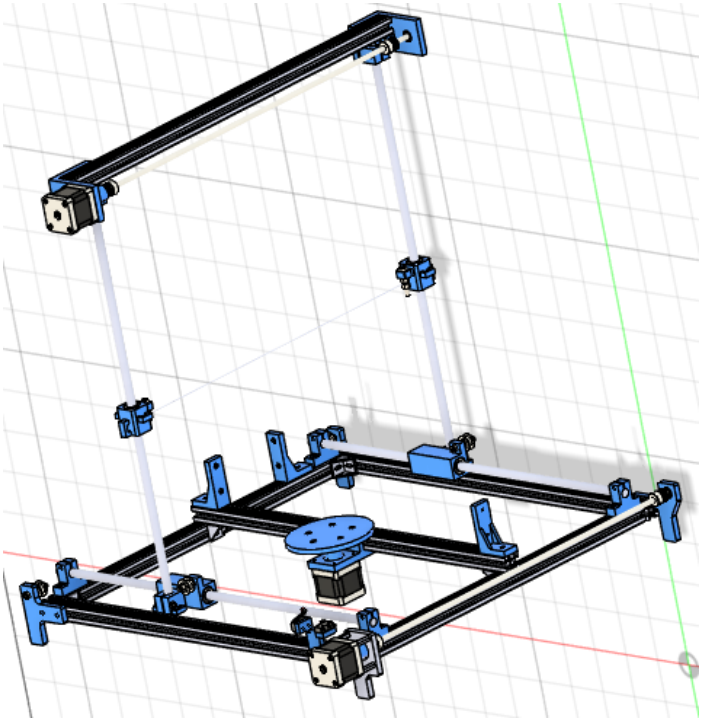


Figure 1

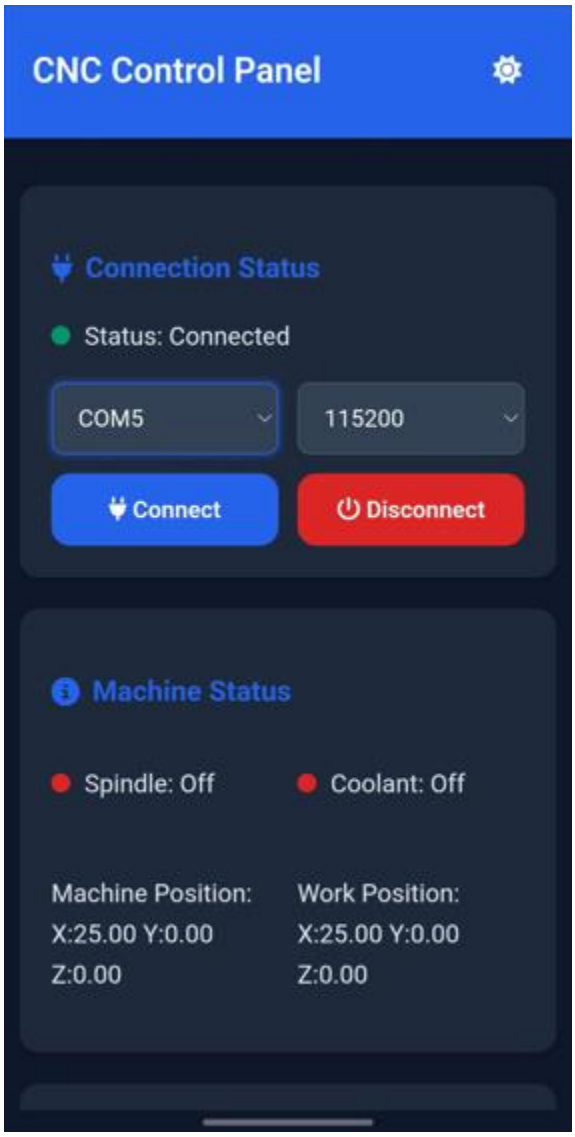
3.4 Control and Automation

- The Raspberry Pi hosts a dual-mode control interface: a direct desktop GUI and a remote web application.
- In the local GUI mode, users can directly jog the machine, set the zero position, and run jobs using the attached touchscreen.

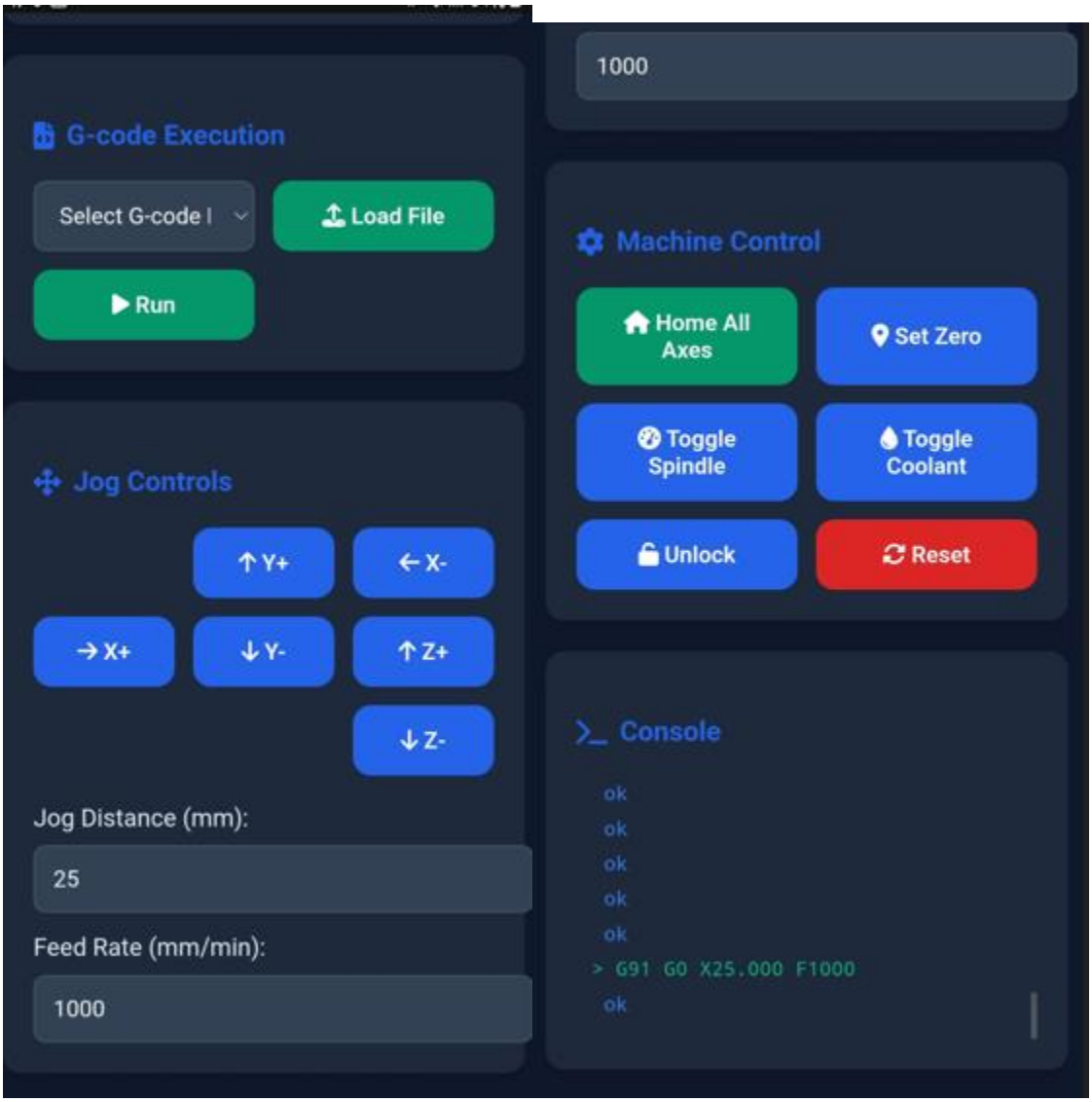
- In the remote mode, users control the machine over Wi-Fi via a responsive web interface. They can upload G-code files, start/pause/stop jobs, jog the machine, and monitor the machine's status and position in real-time.
- The Raspberry Pi acts as the main brain, parsing G-code and sending movement commands (G0, G1, etc.) to the Arduino Uno.
- The Arduino Uno, running GRBL firmware, translates these high-level commands into precise step and direction signals for the stepper motor drivers controlling the X, Y, and Z axes.
- A dedicated DC-DC converter regulates the current to the nichrome wire, allowing for adjustable temperature control to ensure clean cuts through different foam types and densities. This heating element is controlled via a digital pin on the Arduino (e.g., M3/M5 commands).
- End-stop switches on each axis are used for homing ($H command) to establish a known reference position, ensuring repeatable accuracy.

3.5 Programming and Circuit Integration

- The Arduino Uno was flashed with the open-source GRBL firmware, which is optimized for CNC control and interprets standard G-code commands.
- The Raspberry Pi was programmed in Python. The core application integrates a Flask web server for the remote interface and a PyQt5 desktop application for the local GUI. Both interfaces communicate with the same backend control logic.
- Serial communication (UART) over USB was established between the Raspberry Pi and the Arduino Uno. A robust communication protocol was implemented to handle command queuing, wait for ok response, and manage timeouts to prevent the machine from stalling.
- The web interface was developed using HTML, CSS, and JavaScript to provide a modern, mobile-friendly dashboard for control and monitoring.
- The electrical system was integrated into a compact unit, with careful attention to power distribution, separating the high-current motor supply from the logic-level signals to the Arduino and Pi to ensure stable operation.


CNC Control Panel
Connection Status
Status: Connected
COM5
115200
Connect
Disconnect
Machine Status
Spindle: Off
Coolant: Off
Machine Position:
X:25.00 Y:0.00
Z:0.00
Work Position:
X:25.00 Y:0.00
Z:0.00

.....



3.6 Component List and Description

The system is built using the following key hardware components:

- **Arduino Uno**

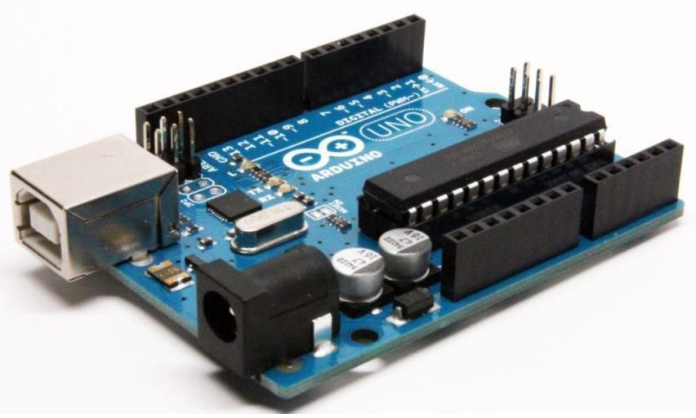


Figure 2

- **Raspberry Pi 4**

Figure 3

- **Nema 17 stepper motor**

Figure 4

- **A4988 stepper motor driver**

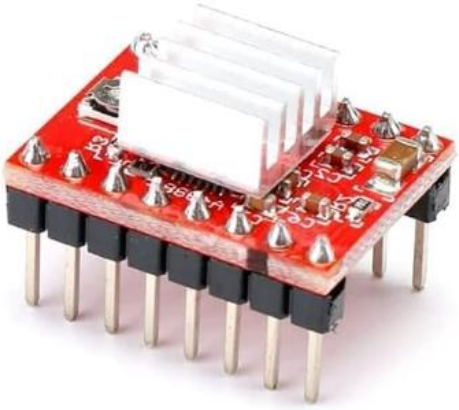


Figure 5

- **DC to DC converter**

Figure 6

- **Power Supply**

Figure 7

- **7inch Touch Screen**

Figure 8

- **Limit Switch**

Figure 9

4 Results and Discussion

4.1 The Design

The mechanical design prioritized rigidity, accuracy, and ease of assembly to ensure precise movement of the cutting tool. The frame was constructed using a combination of materials chosen for their specific properties.

The core structure is built from 20x20mm aluminum extrusions, selected for their excellent strength-to-weight ratio, modularity, and ease of customization. The linear motion systems utilize hardened steel precision rods paired with linear ball bearings to provide smooth, low-friction movement along all three axes (X, Y, and Z). Drive power is transmitted via threaded steel leadscrews with anti-backlash nuts to eliminate play and ensure precise positional accuracy.

Critical components such as motor mounts, couplers, and the wire tensioning system were 3D printed using PLA+ filament. This allowed for rapid prototyping, custom geometric complexity, and lightweight parts. To protect the sensitive electronic components from dust and accidental damage, a dedicated laser-cut acrylic enclosure was designed and fabricated. This enclosure houses the Raspberry Pi, Arduino, power supplies, and motor drivers, providing both protection and visibility for status indicators.

This combination of materials—aluminum for the primary structure, steel for precision movement, and 3D-printed/acrylic for custom components—resulted in a rigid, accurate, and reliable platform for CNC foam cutting.

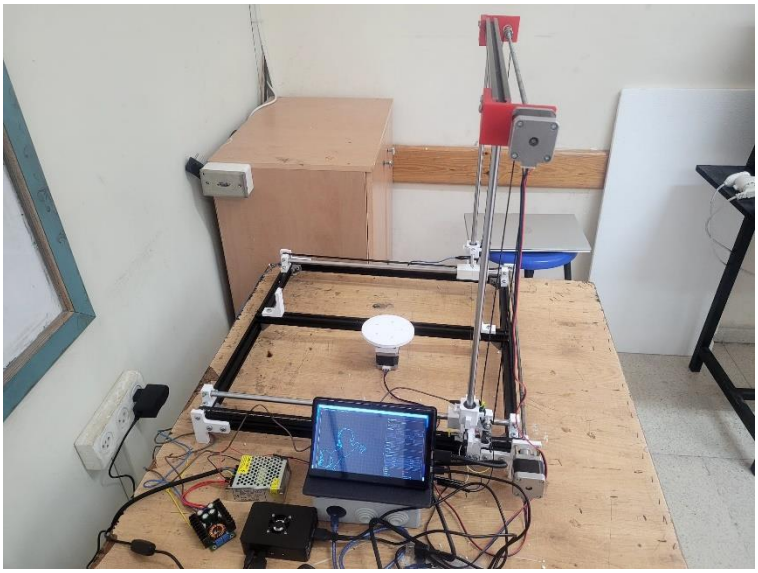


Figure 10

4.2 Final Project

The final outcome of the ScuptoFoam:

1. A robust 3-axis gantry system constructed from linear rods and threaded leadscrews, providing precise movement along the X, Y, and Z axes.
2. A heated nichrome wire cutting tool with adjustable temperature control via a DC-DC converter, enabling clean and smooth cuts through various foam densities.
3. An Arduino Uno microcontroller running GRBL firmware, responsible for precise real-time control of the stepper motors and the heating element.
4. A Raspberry Pi single-board computer serving as a standalone interface, eliminating the need for an external laptop or PC.
5. A dual-mode control system consisting of a local PyQt5 desktop GUI for direct control and a Flask-based web application for remote operation over Wi-Fi.
6. An integrated control panel featuring an attached display and touchscreen for direct user interaction with the machine.
7. End-stop switches on all three axes for accurate homing and establishing a reliable machine coordinate system.
8. A complete and portable enclosure housing the electronics, power supplies, and safety features, creating a single, cohesive unit ready for operation.

4.3 Project Outcomes

The project successfully met its key objectives:

- It provides an effective and affordable alternative to both manual foam cutting and expensive industrial CNC machines, making precision fabrication accessible for education, prototyping, and small-scale production.
- It achieved a high degree of cutting accuracy and repeatability for both 2D shapes and 3D contours, validating the mechanical design and control system.
- Operated as a fully standalone system with low power consumption, eliminating the dependency on a traditional computer or laptop for operation.
- Enabled intuitive user control through a dual-interface system: a direct touchscreen GUI and a responsive web application accessible from any mobile device or computer on the same network.
- Successfully integrated and demonstrated the interoperability of key components—including the Raspberry Pi, Arduino Uno, stepper motors, and heated cutting tool—into a single, functional prototype.
- Validated the potential of this design as a foundational platform for future enhancements, such as multi-material cutting or automated bed-leveling.

5 Conclusions and Recommendation

5.1 Conclusions

This project successfully demonstrated the development of a portable, standalone CNC foam cutting machine that offers a precise, affordable, and accessible alternative to both manual methods and industrial CNC systems. Through the integration of embedded systems, mechanical design, and software development, the prototype effectively bridges the gap between DIY projects and professional equipment.

The system fulfilled its design goals by achieving:

- Precision Cutting Performance: The machine successfully produced accurate and clean cuts in both 2D shapes and 3D contours, validating the effectiveness of the heated nichrome wire and the precise motion control provided by the stepper motors and GRBL firmware.

- Robust System Integration: The reliable serial communication between the Raspberry Pi and Arduino Uno, coupled with the dual-interface control system, demonstrated a robust and functional integration of software and hardware components.

- Operational Independence and Efficiency: By operating as a fully standalone unit with an integrated touchscreen and low-power components, the system eliminated the dependency on an external computer, enhancing its portability and ease of use in various settings like workshops and classrooms.

- User-Friendly Control: The dual-interface system, featuring both a local GUI and a responsive web application, provided an intuitive and flexible control experience, making the machine accessible to users with varying levels of technical expertise.

- Educational and Prototyping Value: The project's modular design, use of open-source software (GRBL), and cost-effective hardware make it an excellent educational tool for learning about CNC technology, robotics, and automation, as well as a valuable asset for rapid prototyping.

Furthermore, the successful convergence of mechanical, electronic, and software elements into a cohesive and functional unit validates the interdisciplinary engineering approach taken throughout the development process. This project serves as a strong foundation for future enhancements and scaling, demonstrating the potential for embedded systems to create sophisticated yet accessible manufacturing solutions.

5.2 Future Work & Recommendations

The modular and scalable design of this CNC platform provides a solid foundation for numerous advanced developments. Future work will focus on expanding the machine's capabilities, intelligence, and autonomy:

- **Extend Material Compatibility**: The system can be modified to cut plastics and lightweight wood. This would involve replacing the heated wire with a high-speed rotary spindle or laser module. Software updates would be

required to manage different feed rates, power levels, and toolpaths optimized for these denser materials, transforming the machine into a versatile multi-material fabrication tool.

- **Integrate a Camera Vision System**: Implementing a camera would significantly enhance functionality. This system could be used for:
    - Automated Bed Calibration: Creating a height map of the material surface to automatically adjust the Z-axis and ensure perfect cutting depth across the entire bed.
    - Job Monitoring and Verification: Providing a live feed to the web interface for remote supervision and allowing the system to halt operations if an error (like a wire break) is detected.
    - Precise Material Alignment: Using fiducial markers to align the physical material with the digital design, ensuring cuts are made in the exact intended location.
- **Implement Automated Material Handling**: To increase throughput and enable continuous operation, a simple conveyor belt or an automated bed shuttle system could be integrated. This would allow for unloading a finished piece and loading a new raw material block automatically or with minimal user intervention, making the machine suitable for small-batch production.

6 References

[1] Raspberry Pi Foundation. (n.d.). Raspberry Pi Documentation. https://www.raspberrypi.com/documentation/

Official documentation for the Raspberry Pi hardware and operating system, used for the standalone interface.

[2] Arduino LLC. (n.d.). Arduino Uno Rev3 Datasheet. https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf

Technical specifications for the Arduino Uno microcontroller, the core controller for motion and tool actuation.

[3] Flask Development Team. (n.d.). Flask Web Development Documentation. https://flask.palletsprojects.com/

Documentation for the Flask web framework, used to create the remote control web server on the Raspberry Pi.

[4] NEMA Stepper Motor. (n.d.). NEMA 17 Stepper Motor Datasheet. https://www.makerguides.com/wp-content/uploads/2019/04/17HS4401S-datasheet.pdf

Representative datasheet for the NEMA 17 stepper motors used on the X, Y, and Z axes.

7 Appendix A: API Documentation and Command Reference

A.1 Serial Command Reference

The Arduino Uno accepts a core set of G-code commands via serial. The Raspberry Pi sends these commands to execute jobs.

A.1.1 Basic Motion Commands

Rapid Positioning (G0)

```
; Rapid move to X10, Y15, Z5 at maximum speed
G0 X10 Y15 Z5
```

Purpose: Quickly move the toolhead to a new position without cutting.

Linear Interpolation Motion (G1) – Cutting Move

```
; Cut a straight line to X50 Y0 at a feed rate of 1000 mm/min
G1 X50 Y0 F1000

; Perform a vertical cut downward
G1 Z-10 F500
```

Purpose: Move the toolhead in a straight line while the cutting wire is active.

A.1.2 Program Flow & Unit Commands

Set Units to Millimeters (G21)

```
G21 ; All coordinates are in millimeters
```

Set to Absolute Positioning (G90)

```
G90 ; All coordinates are relative to machine origin (0,0,0)
```

Set to Relative Positioning (G91)

```
G91 ; All coordinates are relative to the current position
G1 X10 ; Move 10mm to the right from current position
```

A.1.3 Tool (Spindle) Control Commands

Turn On Heated Wire (M3)

```
M3 S255 ; Turn on wire heater at full power (255)
M3 S128 ; Turn on wire heater at half power (128)
```

Turn Off Heated Wire (M5)

```
M5 ; Turn off wire heater
```

A.1.4 Homing and System Commands

Homing Cycle ($H)

```
$H ; Home all axes. Machine will move until it triggers limit switches.
```

Unlock Machine ($X)

```
$X ; Clear the 'Alarm' state, often needed after a reset or error.
```

View Settings ($$)

```
$$ ; List all current GRBL configuration settings
```

View Status ('?')

```
? ; Request a real-time status report (position, state, etc.)
```

Typical Response:
`<Idle|Run|Hold|Alarm,MPos:10.000,0.000,5.000, WPos:0.000,0.000,0.000>`

A.2 Raspberry Pi Flask HTTP API

The Flask server on the Raspberry Pi provides a RESTful API for remote control over Wi-Fi.

Base URL: `http://[RASPBERRY_PI_IP]:5000/api/`

A.2.1 Machine Control Endpoints

POST /api/connect – Connect to the Arduino/GRBL controller.

```
// Request Body
{
  "port": "/dev/ttyACM0",
```

```
  "baudrate": 115200
}

// Success Response
{
  "success": true,
  "message": "Connected to /dev/ttyACM0 at 115200 baud"
}
```

POST /api/jog – Jog the machine a specified distance.

```
// Request Body
{
  "axis": "X",
  "distance": 5.0,
  "feedrate": 1000
}

// Success Response
{
  "success": true,
  "new_position": {
    "X": 15.24,
    "Y": 0.0,
    "Z": 0.0
  }
}
```

POST /api/home – Home all machine axes.

```
// Request Body: {} (Empty)

// Success Response
{
  "success": true,
  "message": "Homing cycle complete"
}
```

POST /api/zero – Set the current work position to zero.

```
// Request Body: {} (Empty)

// Success Response
{
  "success": true,
  "message": "Work position zero set"
}
```

A.2.2 Job Management Endpoints

POST /api/job/start – Start executing the loaded G-code file.

```
// Success Response
{
  "success": true,
  "message": "Job started",
  "total_lines": 150
}
```

POST /api/job/pause – Pause the currently running job.

```
// Success Response
{
  "success": true,
  "message": "Job paused",
  "paused_at_line": 75
}
```

POST /api/job/stop – Stop the current job and reset the buffer.

```
// Success Response
{
  "success": true,
  "message": "Job stopped"
}
```

POST /api/upload – Upload a G-code file to the Raspberry Pi.

```
<!-- Form Data -->
<!-- Input: file (file upload) -->
```

A.2.3 Status Endpoints

GET /api/status – Get the current machine status.

```
// Response
{
  "state": "Idle",
  "machine_position": {
    "X": 0.0,
    "Y": 0.0,
    "Z": 0.0
  },
  "work_position": {
    "X": 0.0,
```

```
    "Y": 0.0,
    "Z": 0.0
  },
  "buffer_state": {
    "available_blocks": 15,
    "rx_bytes": 128
  }
}
```

A.3 WebSocket Events (Real-time Updates)

The web interface uses a **WebSocket connection** for real-time data.

Connection URL: `ws://[RASPBERRY_PI_IP]:5000/ws`

A.3.1 Server to Client Events

Status Update

```
{
  "event": "status",
  "data": {
    "state": "Run",
    "line_current": 42,
    "line_total": 150
  }
}
```

Position Update

```
{
  "event": "position",
  "data": {
    "X": 25.43,
    "Y": 10.12,
    "Z": 0.0
  }
}
```

Console Output

```
{
  "event": "console",
  "data": {
    "message": "> G1 X10 Y20 F1000",
    "type": "tx" // Can be 'tx', 'rx', or 'error'
  }
}
```

Job Finished

```
{
  "event": "job_finished",
  "data": {
    "reason": "completed", // or "stopped", "error"
    "total_lines": 150,
    "total_time": "00:05:23"
  }
}
```

A.3.2 Client to Server Events

Emergency Stop

```
{
  "command": "emergency_stop"
}
```

Feed Hold / Resume

```
{
  "command": "feed_hold"
}
{
  "command": "cycle_resume"
}
```

A.4 Error Codes and Troubleshooting

Error Code	Message	Probable Cause & Solution
ERR_001	Serial port not found	Arduino is disconnected. Check USB cable and power.
ERR_102	G-code word not recognized	Unsupported G-code command sent. Check file syntax.
ERR_202	Homing fail. Please reset	Axis could not home. Check limit switch wiring.
ERR_305	Job start failed: No file loaded	Attempted to run a job without loading a G-code file.
ERR_408	Jog rejected: Machine is running	Cannot jog while a job is in progress.

WS_ERR	WebSocket connection failed	Raspberry Pi Flask server is not running.