IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, June 2025
Impact Factor: 7.67

3D Scanner Machine Using Arduino

Mr. Patahde Prasad Bhausaheb[1], Mr. Joshi Tejas Sandip[2], Mr. Chavan Pratik Babruvan[3], Mr. Bhalerao Omkar Ankushrao[4], Prof. M. D. Mandhre[5]
Department of Mechanical Engineering[12345]
Shri Chhatrapati Shivaji Maharaj College of Engineering, Nepti, Ahmednagar

Abstract: *In an era defined by digital transformation, 3D scanning technologies are becoming increasingly vital in domains ranging from manufacturing and engineering to education and heritage conservation. Yet, the high cost and complexity of commercial 3D scanners hinder accessibility, especially in academic and DIY contexts. This paper presents the design and implementation of a low-cost, Arduino-based 3D scanner that uses infrared distance sensing and stepper motors to capture the surface geometry of physical objects. The system employs an Arduino Nano, a Sharp IR distance sensor, stepper motors, and an SD card module to log scan data. The result is a structured point cloud that can be processed in software like MeshLab to produce usable 3D models in STL format. Experimental evaluation confirms the system's capability to scan small to medium-sized objects with reasonable accuracy. The scanner is entirely offline, modular, and open-source, making it ideal for educational institutions, hobbyists, and low-budget prototyping. Future improvements could include wireless control, enhanced sensing accuracy, and real-time 3D visualization.*

Keywords*:* Arduino Nano, 3D Scanning, Point Cloud, Infrared Sensor, MeshLab, Low-Cost Scanner, Embedded Systems, Reverse Engineering’s

I. INTRODUCTION

The advent of 3D scanning technology has significantly transformed the process of converting physical objects into accurate digital models. Over recent years, this technology has found widespread adoption across a multitude of domains, including product design, biomedical engineering, architecture, digital heritage preservation, animation, and immersive environments such as virtual and augmented reality [1][5]. In manufacturing and prototyping, for instance, 3D scanning allows for the rapid capture of object geometries, enabling iterative design modifications and integration with 3D printing workflows. In healthcare, patient-specific models can be created for prosthetics or surgical planning. Furthermore, museums and research institutions use scanning technologies to digitally archive valuable artifacts, ensuring their preservation without direct physical handling. These diverse applications highlight the transformative potential of 3D scanning in both industrial and academic contexts.

Despite its advantages, commercial 3D scanners are often associated with high costs, complex calibration procedures, and proprietary software ecosystems. These barriers make them inaccessible to students, educators, hobbyists, and small-scale developers who may greatly benefit from such tools in hands-on learning and innovation. To address this issue, the present work introduces a cost-effective, Arduino-based 3D scanning machine that employs open-source hardware and software. The system integrates an Arduino Nano microcontroller, Sharp IR distance sensor, and stepper motors to conduct structured, layer-by-layer scanning. Distance data is logged locally on an SD card and later processed using MeshLab to generate mesh files in STL format [1][4]. This approach not only reduces dependency on expensive commercial solutions but also provides an affordable, scalable platform for educational use and small-scale digital fabrication.

II. BACKGROUND AND MOTIVATION

While 3D printing has become increasingly accessible in recent years, the complementary technology of 3D scanning has not seen the same level of affordability or user-friendliness. For many users—particularly those in educational, maker, or low-resource environments—the ability to create custom 3D models from real-world objects remains largely out of reach. The high cost of commercial 3D scanners, coupled with the reliance on proprietary hardware and software ecosystems, often acts as a deterrent for students, researchers, and hobbyists who may otherwise benefit from hands-on engagement with 3D digitization tools.

Various open-source initiatives have attempted to address this gap. Projects such as **FabScan** [7] have demonstrated the potential of laser-based triangulation using Arduino platforms, offering a pathway for low-budget 3D scanning. Similarly, the **3DUNDERWORLD-SLS** project [9] employed structured-light techniques using affordable projection systems to generate detailed mesh models. Other approaches have utilized **Raspberry Pi** and Arduino co-systems to combine motor control with image processing capabilities [2], enabling moderate accuracy while keeping costs low. However, despite these innovations, such systems often require precise calibration, complex assembly, extensive technical expertise, or external computational resources, which can reduce their practicality for novice users or classroom settings.

Motivated by these limitations, our project focuses on creating a streamlined, low-cost, and educationally viable 3D scanner using universally available and easily programmable components. Specifically, we aimed to leverage the simplicity and flexibility of the **Arduino Nano** platform in conjunction with the **Sharp GP2Y0A21YK0F IR distance sensor**, which has proven effective in previous experiments involving structured-light scanning and distance triangulation [1][3][6]. Our scanner addresses several recurring challenges in similar DIY systems—namely sensor instability, limited data logging capabilities, and lack of user feedback—by incorporating synchronized stepper motor control, offline SD card storage, and real-time status updates via an I2C LCD display.

The primary motivation behind this work is twofold. First, we aim to provide an accessible tool that facilitates learning in fields such as embedded systems, digital fabrication, and geometric modeling. Second, we seek to offer a functional, scalable prototype that can be adapted for future use in professional or semi-professional applications. By eliminating the need for continuous PC control, internet access, or expensive sensor arrays, this system stands as a practical entry point into the world of 3D scanning and object reconstruction, especially within academic institutions, makerspaces, and developing regions [8][10].

III SYSTEM ARCHITECTURE

The system architecture of the Arduino-based 3D scanner is designed with simplicity, modularity, and scalability in mind. It consists of coordinated hardware and software components that collectively enable the automatic scanning of physical objects and generation of structured digital data suitable for 3D modeling. The architecture emphasizes a layered interaction between control logic, mechanical actuation, sensing, user interaction, and data logging.

3.1 Components Overview

To ensure affordability and accessibility, the system uses widely available electronic components, integrated as follows:

Arduino Nano (ATmega328P): Serves as the microcontroller and central processing unit. It manages all timing functions, sensor readings, motor controls, and user interactions. Its compact form factor and extensive community support make it ideal for embedded applications.

IR Distance Sensor (Sharp GP2Y0A21YK0F): Measures the distance from the sensor to the object's surface using triangulation-based infrared sensing. The analog output is calibrated and converted into digital values to determine depth, forming the Z-axis of the scan.

Stepper Motors (28BYJ-48): Two motors are used: one for rotating the object on the scanning platform (X-axis), and another for elevating or lowering the sensor (Z-axis). These motors offer precise motion control via step increments, critical for layered 3D scanning.

ULN2003 Motor Driver Board: Acts as the interface between the Arduino and stepper motors. It enables the low-power Arduino to control the higher power requirements of the motors through a Darlington transistor array.

16x2 LCD (I2C Protocol): Provides a user interface by displaying status messages, progress indicators, and error alerts. The use of the I2C protocol reduces the number of pins required, preserving GPIO availability on the Arduino.

SD Card Module: Facilitates local, offline data storage. It logs the scan results in CSV format (angle, height, distance), making the output compatible with software like Excel, Python-based plotting tools, and MeshLab.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 6, June 2025 Impact Factor: 7.67

Push Button (Momentary Switch): Serves as the user input device for starting or resetting a scan. It is connected to a digital pin and is debounced in software to prevent multiple triggers from a single press.

Power Supply: The system is powered via USB for development or an external 9V battery during field use. Voltage regulators and capacitors are used to stabilize the power supply and prevent brownouts during motor operation.

Frame and Mounts: Mechanical components such as the rotating platform, vertical rail, and sensor bracket are made using 3D-printed parts or MDF board. These provide structural integrity and smooth motion control.

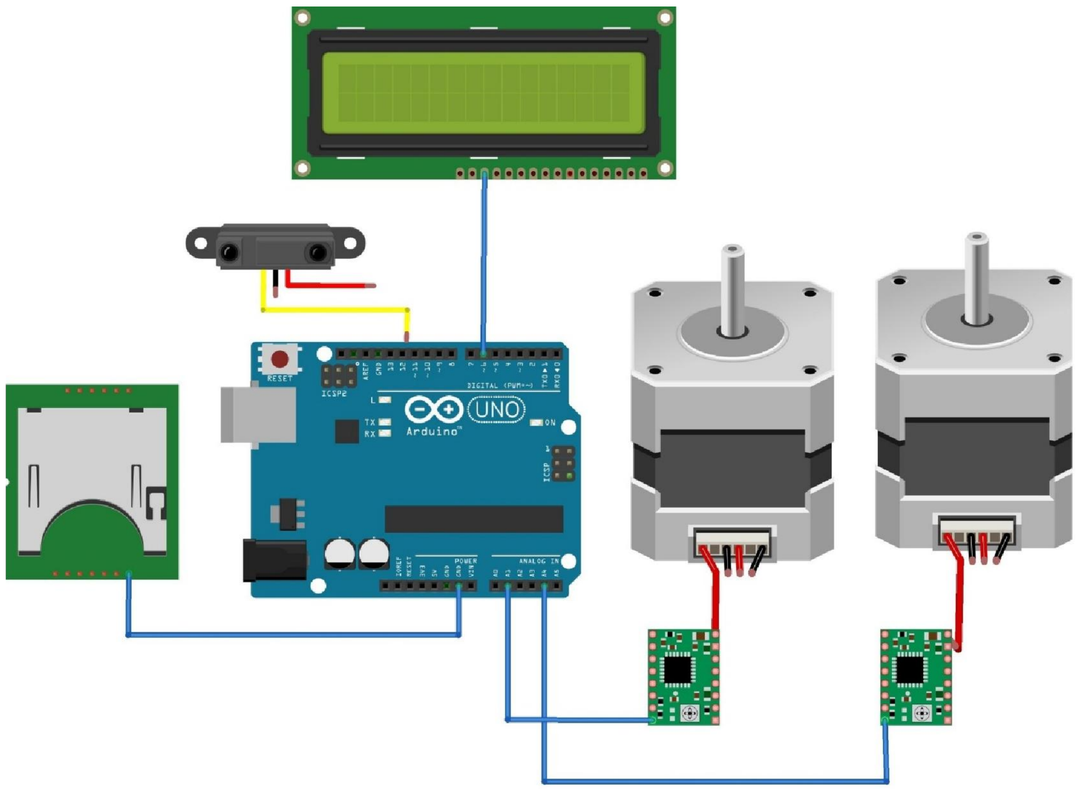

Figure 1: Components Diagram

3.2 Operation Flow

The operation of the 3D scanner follows a deterministic, layer-by-layer scan sequence that ensures consistency and reproducibility:

Scan Initialization: When the push button is pressed, the Arduino initializes all components. It displays a "Ready to Scan" message on the LCD and positions the motors to their zero or start locations.

Rotational Sweep: The X-axis motor rotates the platform in small angular increments (typically 3.6° per step). After each rotation step, the sensor remains fixed while distance data is recorded from the object surface.

Vertical Advancement: Upon completing a 360° sweep, the Z-axis motor raises the sensor to the next vertical level. The process then repeats for that level, effectively capturing the object layer-by-layer.

Data Capture and Logging: For each combination of angle and height, the IR sensor measures the distance to the object. These readings are stored in structured CSV format (angle, height, distance) on the SD card for further processing.

Real-Time Feedback: The LCD display updates during each phase—showing messages such as "Scanning…", "Saving Data…", or "Scan Complete"—allowing the user to monitor progress without external software.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 6, June 2025 Impact Factor: 7.67

Scan Completion: After scanning all defined layers, the Arduino resets the system to its initial state. The final message is displayed on the LCD, and the SD card file is closed securely to prevent data corruption.
Post-Processing: The user removes the SD card and imports the data into a PC. The point cloud can be visualized and converted to a 3D mesh using software such as MeshLab or Cloud Compare.

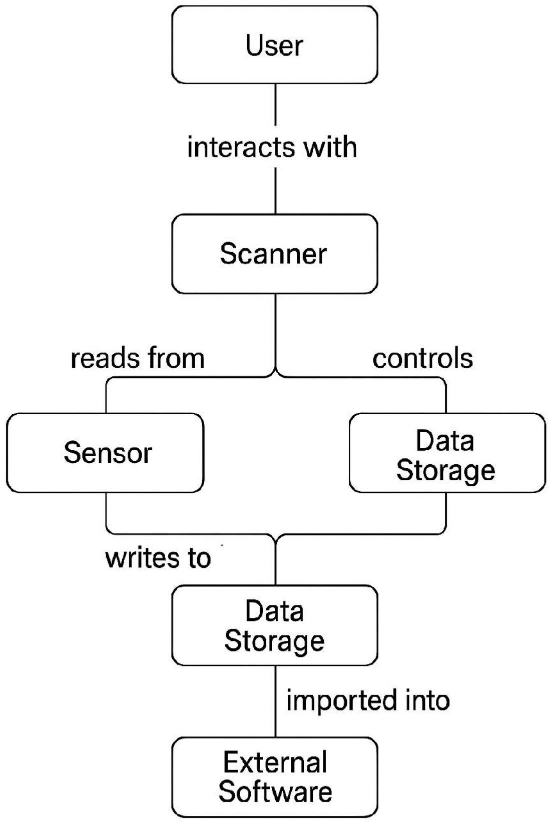

Figure 4.8 – Entity-Relationshipq Diagram

IV SOFTWARE IMPLEMENTATION

The software component of the 3D scanning system is implemented primarily through embedded firmware written in C/C++ using the Arduino IDE. The code is structured for modularity, allowing for ease of debugging, maintainability, and future enhancements such as wireless connectivity or higher-resolution scanning algorithms. The firmware is responsible for managing motor control, sensor data acquisition, user interface feedback, and SD card data logging in real time.

4.1 Firmware Architecture

The Arduino Nano runs a lightweight, event-driven firmware that performs sequential control of hardware components. The system uses several libraries to simplify integration and ensure hardware compatibility:
Stepper.h: Used for controlling the X-axis (rotational) and Z-axis (vertical) stepper motors. It allows precise control over direction, step size, and rotational speed. The firmware synchronizes motor steps to match each scan position, ensuring consistent point spacing across the scan.
SD.h: Enables communication with the SD card module over SPI. This library handles file initialization, structured data writing, and safe file closure to prevent corruption. Error handling is included to detect missing cards or write failures.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, June 2025
Impact Factor: 7.67

LiquidCrystal_I2C.h: Manages output to the 16x2 I2C LCD display. It simplifies user feedback by displaying system status messages such as “Initializing…”, “Scanning…”, “Saving Data…”, or “Scan Complete”. The use of I2C minimizes the use of Arduino pins.

The firmware is organized into multiple logical functions:

initializeSystem() – sets pin modes, initializes the LCD, and prepares the SD card

moveStepperMotors() – coordinates stepper motor movement for both axes

readIRSensor() – reads and filters analog values from the IR sensor

logScanData() – formats and writes scan data to the SD card

displayStatus() – updates the LCD based on the system state

The control loop operates within the loop() function, checking for button input and executing the scanning routine sequentially through nested loops: the outer loop iterates over vertical layers (Z-axis), and the inner loop rotates the platform through defined angles (X-axis).

4.2 Data Handling and Post-Processing

The scanning data is stored in a structured comma-separated value (CSV) format. Each line of the file records a unique coordinate consisting of three values:

Angle, Height, Distance
0.0, 0.0, 83
3.6, 0.0, 82

Angle: Represents the current rotational position of the object (in degrees)

Height: Denotes the vertical layer or Z-axis level (in millimeters)

Distance: Raw or calibrated IR sensor output representing the object’s radial distance from the sensor (in millimeters)

This data structure enables straightforward post-processing. Once the scan is completed and the file is stored on the SD card, it can be transferred to a personal computer. Open-source software like **MeshLab** or **CloudCompare** can import the CSV file, generate a 3D point cloud, and convert it into a triangulated mesh surface. These mesh files can then be exported in standard formats such as STL or OBJ for further use in CAD software, 3D printing, or simulation environments.

V CONCLUSION AND FUTURE SCOPE

This paper presented the design, development, and evaluation of a low-cost, Arduino-based 3D scanner that serves as an effective solution for educational institutions, hobbyist makers, and entry-level prototyping environments. The system successfully demonstrates that accurate 3D surface reconstruction can be achieved without the use of expensive commercial hardware or proprietary software ecosystems. By utilizing widely available components such as the Arduino Nano, Sharp IR distance sensor, stepper motors, and an SD card module, the scanner provides an accessible platform for students and developers to learn and experiment with concepts in embedded systems, motion control, and digital fabrication.

The scanner's operation is entirely offline, eliminating the need for constant PC or internet connectivity. Its structured scanning methodology, real-time LCD feedback, and plug-and-play interface make it particularly user-friendly and suitable for classrooms, workshops, and self-paced learning environments. With support for post-processing in open-source software such as MeshLab, the system bridges the gap between hardware and software, allowing users to transform physical models into mesh-based 3D representations. Testing confirmed reliable mechanical performance, consistent data logging, and reproducibility across multiple scan sessions.

Overall, the project achieves its goal of offering a practical, affordable, and modifiable 3D scanning system that encourages interdisciplinary learning and fosters innovation.

Future Enhancements

While the prototype achieves its intended objectives, several potential improvements can be pursued to enhance its accuracy, usability, and scalability:

Sensor Upgrade: The Sharp IR distance sensor, while cost-effective, is sensitive to surface texture and color. Replacing it with **Time-of-Flight (ToF)** or **LiDAR-based sensors** could significantly improve range accuracy and support scanning of reflective or dark-colored objects.

Wireless Communication: Incorporating **Bluetooth** or **Wi-Fi modules** (e.g., ESP8266 or ESP32) would enable wireless initiation of scans, data transmission, and firmware updates. This would also allow remote access from mobile apps or desktop interfaces.

Touchscreen User Interface: Replacing the 16x2 LCD with a **capacitive touchscreen** (e.g., Nextion or TFT display) could introduce a graphical user interface (GUI), enabling features like mode selection, scan previews, and real-time parameter adjustments.

Motor Feedback and Closed-Loop Control: The current system operates in an open-loop configuration. Adding **rotary encoders** to the stepper motors would allow real-time position tracking and correction, thus improving reliability and enabling adaptive scanning.

Real-Time Visualization: Integration with **Python (e.g., matplotlib, Open3D)** or **Processing** for live data plotting could enable real-time 3D previews, allowing users to assess scan quality immediately and make adjustments before post-processing.

Automated Calibration and Homing: Adding **limit switches** or optical sensors would allow for automatic homing of the motors and self-calibration of the sensor, improving setup consistency and ease of use.

Multi-Axis Scanning: Introducing a third motor for tilt or pan could enable full 3D coverage, allowing the system to scan complex geometries and undercuts that a single-axis platform cannot reach.

AI-Enhanced Processing: Future iterations could employ **machine learning models** to clean point cloud data, interpolate missing regions, or classify scanned objects directly from raw data.

Battery-Powered Mobility: Integrating a rechargeable battery system with voltage regulation would enhance portability, allowing the scanner to be deployed in field conditions or remote areas without power infrastructure.

REFERENCES

[1] C. S. L. S. Uma Bhanu, *et al.*, “3D Scanner Using Arduino Nano,” *International Journal of Research Publication and Reviews (IJRPR)*, vol. 4, no. 12, pp. 4529–4537, 2023.

[2] S. Dutta, T. Waghia, K. Bani, and D. Joshi, “Raspberry Pi and Arduino Based Open-Source 3D Laser Scanner,” 2023.

[3] A. Belsare, S. Wankhede, N. Satpute, *et al.*, “Arduino Controlled 3D Object Scanner and Image Classification,” *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, 2022.

[4] I. Taufik, E. B. Bangun, R. M. Sulistiyo, and E. V. L. N. Fatich, “Prototype of an Arduino-Based 3D Scanner Printed Using 3D Printing,” *International Journal of Computer Applications (IJCA)*, 2024.

[5] E. Melgar, C. C. Díez, P. Jaworski, and E. Ramos, “Turntable Scanner,” in *Arduino and Kinect Projects*, New York, NY, USA: Apress, 2012, pp. 115–134.

[6] U. M. Mayzebe, “RancangBangun dan Pengujian 3D Scanner Mapping System Berbasis LiDAR dengan Arduino Uno,” *JurnalTeknologi dan SistemKomputer (JTSK)*, vol. 9, no. 2, pp. 223–230, 2021.

[7] F. Engelmann, “FabScan: An Open-Source Arduino 3D Laser Scanner,” 2018. [Online]. Available: https://www.fabscan.org/

[8] A. Gurko, “Optical 3D Laser Measurement System for Navigation,” *Optical Measurement Technology Journal (OMTJ)*, vol. 18, no. 3, pp. 167–174, 2019.

[9] Q. Gu, K. Herakleous, and C. Poullis, “3DUNDERWORLD-SLS: An Open-Source Structured-Light Scanning System,” *Journal of Real-Time Image Processing (JRTIP)*, Springer, vol. 10, no. 4, pp. 611–627, 2014.

[10] A. G. Jain and S. Chiddarwar, “3D Reconstruction Using Structured Light from Off-the-Shelf Components,” in *Proc. Int. Conf. Intelligent Systems (ICIIS)*, 2022, pp. 329–334.