© May 2026 | IJIRT | Volume 12 Issue 12 | ISSN: 2349-6002

HARI: A Low-Cost AI -Powered Interactive Mobile Robot for Autonomous Campus Navigation & Assistance Using Raspberry Pi and OpenCV

Tanish D. Pawar[1], Aditya R. Mahajan[2], Pratham S. Kamble[3], Tanuja A. Gavade[4], Chinmayee C. Sinkar[5], Murad M. Kureshi[6], Shivtej P. Randive[7], Onkar C. Sutar[8], Harsh P. Pathrut[9], Shital P. Chavan[10]

[1,2,3,4,5,6,7,8,9]*Mechanical Engineering Department, D. Y. Patil College of Engineering & Technology, Kasaba Bawada, Kolhapur*

[10]*Projct Guide Mechanical Engineering Department, D. Y. Patil College of Engineering & Technology, Kasaba Bawada, Kolhapur*

***Abstract*— In hospitals, transportation hubs, and educational campuses, service robotics has advanced from controlled industrial environments to public-facing human-centric applications. This work presents HARI (Human Assistance Robotic Interface), a multi-functional mobile robotic platform designed to assist navigation, interaction, and service delivery within university campuses. The system integrates facial recognition, touch-screen-based interaction, autonomous line-following navigation, and localized data services into a unified architecture. Computing is distributed between a Raspberry Pi 4B for high-level AI and GUI processing and an Arduino Mega 2560 for deterministic real-time motor and sensor control, communicating over USB Serial. The navigation subsystem employs PID-controlled lines following augmented with the LSRB decision algorithm and RFID-based positional checkpoints, achieving a lateral deviation below ±2 cm. Face recognition achieves 90–95% accuracy using a dib 128-dimension encoding pipeline. Delivery trials over 40 runs yielded a 95% task-completion rate. Fiber composite panels reduce chassis weight while maintaining structural integrity, keeping total fabrication cost low. Experimental results confirm stable, reliable performance across all core functions, establishing HARI as a scalable, modular, and economically viable solution for smart-campus deployment.**

***Index Terms*— Service robotics, human–robot interaction, embedded systems, face recognition, autonomous navigation, Raspberry Pi, Arduino, OpenCV, PyQt5, PID control, smart campus**

I. INTRODUCTION

QA. Background

Assistance robotic systems have advanced significantly, with rapid growth occurring from industrial automation into human-centric service applications [2]. Airports, hospitals, and retail spaces now deploy guidance, logistics, and interaction systems as a standard feature Developments in embedded computing platforms, low-cost sensors, and open-source computer vision frameworks have been central to this transition [6].Core advances in mobile robotics particularly in localization, navigation algorithms, and lightweight chassis design have enabled scalable implementations even in resource-constrained settings .Modern educational institutions can leverage these technologies to build custom robotic platforms
tailored to campus-specific requirements without prohibitive investment

B. Problem Statement

Educational institutions face persistent challenges in navigation assistance and information dissemination. New students and visitors frequently struggle to locate destinations; meanwhile, document and small-item transfers remain manual and error-prone. Existing solutions tend to address only a single function delivery or navigation making dedicated procurement expensive. This fragmentation motivates the design of an integrated, multi-functional platform [4].

C. Proposed Solution
The HARI system is designed as a multi-functional robotic platform integration:

a. Autonomous line-following and RFID-guided campus navigation
b. Interactive information retrieval via a local SQLite database
c. Face recognition-based attendance logging
d. Small-scale delivery services with QR-verified destination confirmation

A key design goal was fully offline operation: all computation and vision tasks executed on local hardware, eliminating dependency on network connectivity and ensuring continuous availability [6].

D. Contributions
The principal contributions of this work are:

a. A multi-functional AI-driven service robot purpose-built for campus environments
b. A touch-based PyQt5 graphical interface requiring zero user training
c. Real-time face recognition using OpenCV and dlib
d. A dual-controller architecture (Raspberry Pi 4B + Arduino Mega 2560) with USB Serial communication
e. Low-cost fiber composite chassis construction reducing weight and motor torque demand
f. Experimental validation across all modules in real-world indoor scenarios

II. TRETURE REVIEW

The design of HARI is informed by prior research across robot navigation, human–robot interaction, computer vision, and low-cost embedded systems.
Techniques For localization and mapping, well-known robots like SLAM employ probabilities medals and mathematics. Nevertheless, these methods are usually not suitable for low-cost embedded deployments same they require high-performance processing gear Rigidity and lighters are the two main Competing criteria for choosing a chairs material on mobile robot according to Jinasena and Meegama [2]. HARI Choose fiber composite panels over heavier metal alternatives because a lighter chassis lowers torque demand EN drive motors and extends battery life.
Like contemporary Smartphone interaction, interfaces for human robot interaction must be simple to use and need little learning. This idea is the foundation of HARI's PyQt5 touchscreen and field experiences from similar deployments verify. that institutional staff com uses these systems on their Own with little braining.
OpenCV enables real-time facial recognition on embedded platforms, making offline identity verification feasible at low cost. The dlib-based 128-dimension feature encoding pipeline adopted in HARI is consistent with approaches validated in prior embedded face recognition studies [3].
allows robots to navigate intersections in Structured Indoor environments without requiring a fully pre-mapped layout [4], further validated line following delivery robots in institutional Settings, reporting 8-95% autonomous task completion and stop point detection accuracy [5].
Existing commercial robotic systems demonstrate navigation and obstacle avoidance but lack integration with interactive interfaces, recognition systems, and delivery capabilities. HARI fills this gap by combining all these features into a single, low-cost platform

III. SYSTEM ARCHITECTURE

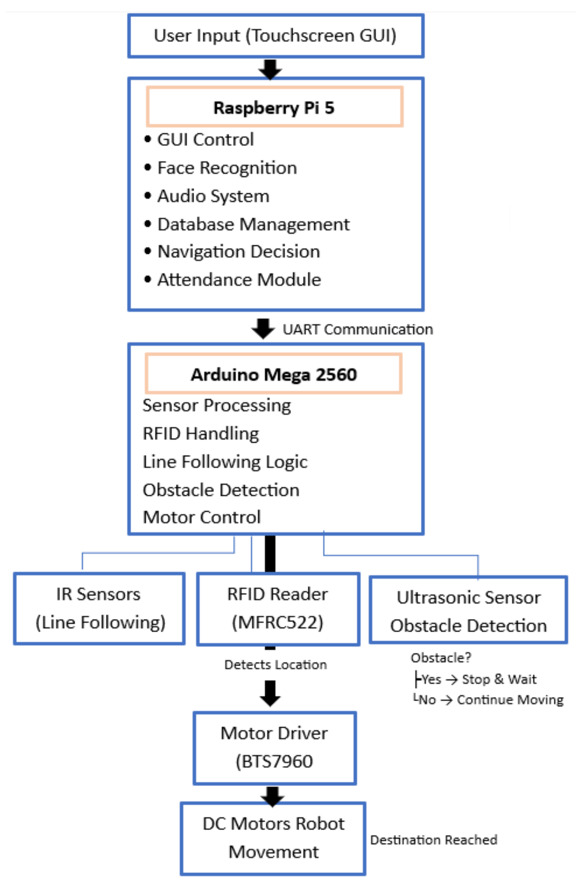


Figure 1: HARI Workflow

A dual-controller design is used by the HARI robot to effectively manage tasks. High-level functions like GUI interaction, facial recognition, audio feedback, database access, and navigation decisions are handled by Raspberry Pi 4B. Real-time hardware control, including sensor processing, RFID-based location detection, line-following logic, obstacle detection, and motor control, is accomplished using an Arduino Mega 2560. UART serial connectivity establishes communication between the two controllers [2]. Path tracking is accomplished by IR sensors, navigation checkpoints are identified by the RFID module, and obstacles are detected while moving by the ultrasonic sensor [4]. The motor driver manages the DC motors to safely steer the robot in the direction of the chosen destination based on sensor feedback [5].

HARI employs a distributed processing architecture with two cooperating controllers:

Raspberry Pi 4B (Master Controller)

Handles AI inference, GUI rendering, SQLite database operations, face recognition, and high-level navigation logic [6]. Running Raspbian OS (64-bit), it executes computationally heavy tasks that would be infeasible on a microcontroller.

Arduino Mega 2560 (Slave Controller)

Controls servo actuation, ultrasonic distance measurement, IR sensor array polling, and real-time motor PWM control [5]. Compared to I2C alternatives, the USB Serial interface for communication with the master controller offers dependable, high-throughput data interchange with easier debugging [6].

Software Modules

Graphical User Interface (GUI) -PyQt5 touch interface [1] Face Recognition Module OpenCV + dlib [3]
Navigation Module PID control [5]
Information Retrieval Module SQLite local database

IV. METHODOLOGY

A. System Initialization

The system boots via Raspbian OS with automated execution of the main application. Diagnostic checks verify motor functionality, camera availability, sensor response, and database connectivity before the GUI is presented to the user [2].

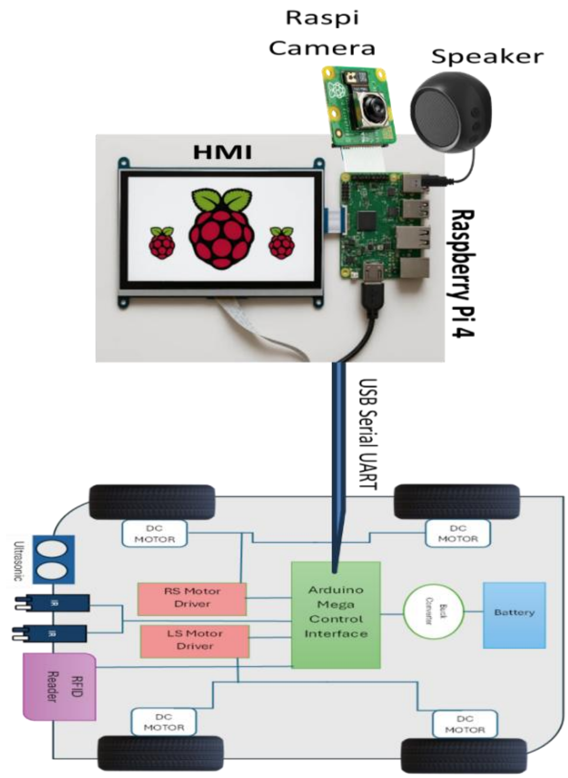


Figure 2: HARI Block Diagram

B. Navigation Subsystem

Graph-based route planning computes optimal paths between campus nodes. Physical movement is governed by three cooperating mechanisms.

PID-regulated line following: The IR sensor array's divergence from the line center is known as the error. To adjust heading, differential motor speed is driven by the control law

$$u(t) = Kp\, e(t) + Ki \int_0^t e(\tau)d\tau + Kd \left(\frac{de(t)}{dt}\right)$$

Equation 1:PID Control Law

Where:

$u(t)$ → Control output

Kp → Proportional gain

Ki → Integral gain

Kd → Derivative gain

$e(t)$ → Error (difference between desired and actual value)

$\int e(\tau)d\tau$ → Accumulated error over time

$\frac{de(t)}{dt}$ → Rate of change of error

When the parameters are adjusted, the robot can manage twists of 90° and more without losing its course, and lateral deviation stays below ±2 cm. For best reflection, infrared sensors are installed with a 3 mm ground clearance. A digital signal is produced by sensors: logic 1 on white surfaces and logic 0 on black or absent-line surfaces. Because black surfaces restrict the IR detection range to 3 cm instead of 24 cm for white, black-on-white line marking is employed throughout to ensure strong contrast [4]

When the line is completely lost, a fault-tolerance mechanism kick in. In more than 90% of situations, the robot successfully recovers the path by stopping and running a scan function [5].

C. Face Recognition Pipeline

The Pi Camera Module 3 is used to take the user's picture before the facial recognition module starts. The dlib model is used to create a 128-dimensional facial feature encoding after OpenCV is used to identify facial areas from the collected frame.

Euclidean distance matching is used to compare the extracted encoding with previously stored face data in the SQLite database. If a legitimate match is found, the display and audio system provide confirmation, and the attendance is automatically logged with a timestamp [3]. The system generates an unknown face alert if there is no match.

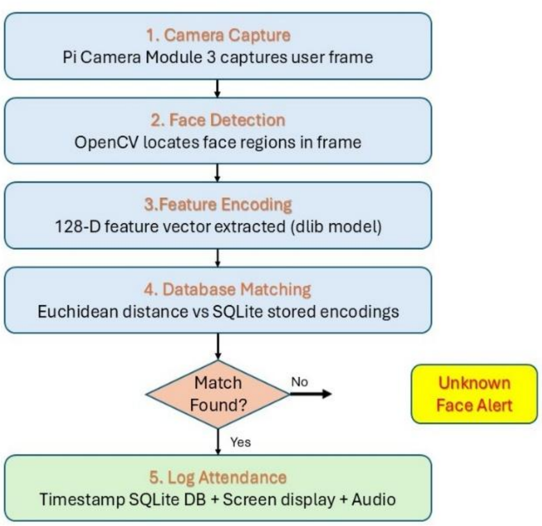


Figure 3: Face Recognition Workflow

D. Information Retrieval

A local SQLite database stores campus maps, room schedules, faculty information, and institutional knowledge. All queries are resolved locally, guaranteeing sub-second response and eliminating network dependency.

E. User Interaction

Four main service options are available through the PyQt5 interface: Attendance System, Campus Information, Delivery Service, and Navigation Assistance. Field reports from similar deployments where institutional staff using the system independently attests to the interface's zero-training usability.

The interactive touchscreen-based user interface of the HARI robot was created using PyQt5 to guarantee easy and intuitive operation for those with no technical expertise. Through big, touch-friendly buttons, users may access department details, faculty information, laboratory information, central amenities, attendance services, timetable schedules, and amusement features in the "Get Help" module, which functions as a centralized information help system.

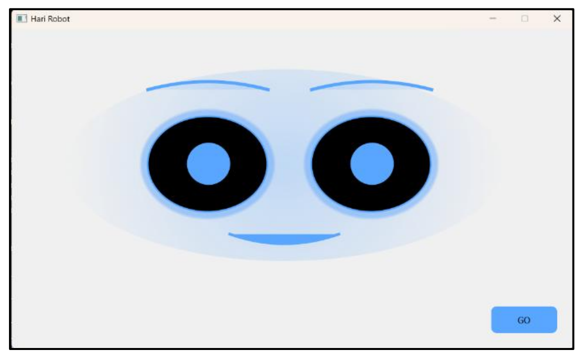


Figure 4: Boot Screen

Home and Back navigation controls are integrated across all screens to maintain consistent user flow and minimize confusion during operation. The interface is designed for quick accessibility in public environments, such as educational institutions, where first-time users may interact with robots without prior training.

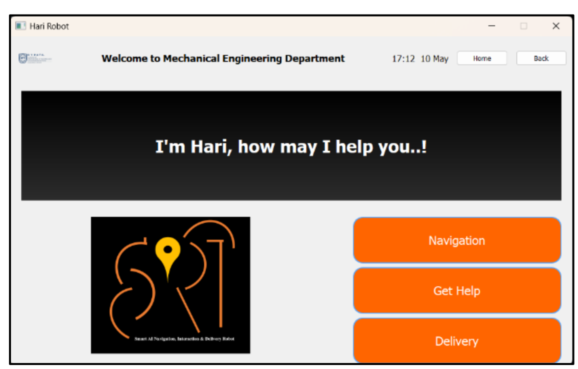


Figure 5: Main menu

A clean layout, high contrast buttons, and simplified navigation structure improves usability and reduces interaction complexity [1]. Usability is enhanced and interaction complexity is decreased with a simple navigation structure, high contrast buttons, and a clean layout. Home and Back navigation controls are linked across all screens to preserve consistent user flow and prevent confusion during operation.

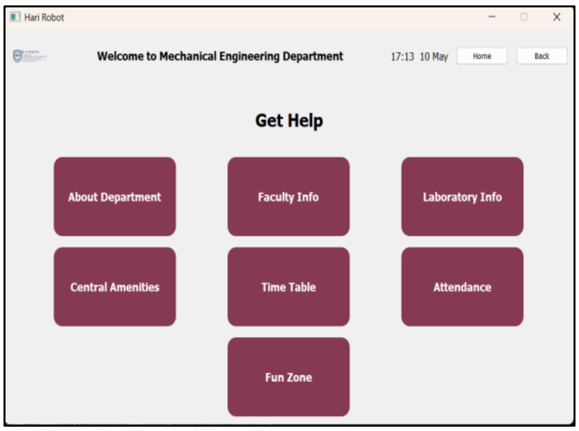


Figure 6: Get Help Module

While the information modules obtain data from a locally saved SQLite database to guarantee quick offline access, the attendance module incorporates camera-based facial recognition for automated attendance reporting. To increase user engagement and show off extra human-robot interaction possibilities, the Fun Zone module offers simple interactive games. In general, responsive interaction, accessibility, and dependable operation in actual campus settings are the main goals of the GUI architecture.

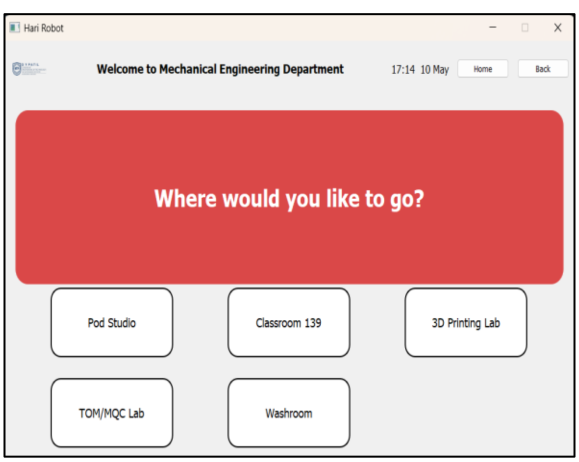


Figure 7: Navigation Module

V. MATERIAL SELECTION

A. Core Hardware Components					
Sr. No	Component	Photo	Model Type	Specification	Reason for Selection
1	Controller		Raspberry Pi 4B	4–8GB RAM	High processing for AI & vision
2	Microcontroller (optional)		Arduino Mega 2560	Atmega 2560	Real-time control
3	Camera		Pi Camera Module 3	5MP	Face recognition
4	Display/ HMI		Touchscreen	1024×600	User interaction

Figure 8: Core Hardware Components

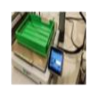
B. Motion & Navigation Components					
Sr. No.	Component	Photo	Model/Type	Specification	Reason for Selection
6	Motors x 4		DC Geared Motors	300 RPM	Balanced speed + torque
7	Wheels x 4		Rubber Wheels	High grip	Smooth movement
8	Chassis Body		4-Wheel Robot Base	Metal + Plastic	Lightweight and Low cost
9	3D Printed Casing and Sensor Holder		Casing	PLA Material	Customization according to use
10	Insulation Tape		Black	Pressure Sensitive Adhesive	Low cost path marking
11	Motor Driver		H- Bridge High- power	BTS7960	High Current Capacity is required to drive the heavy duty motors of the service robot

Figure9: Motion & Navigation Component

C. Sensors					
Sr. No.	Component	Photo	Model/Type	Specification	Reason for Selection
12	Ultrasonic Sensor		US-100 Ultrasonic Sensor	Distance measurement	Obstacle detection
13	IR Sensor x 2		IR Module	Line following	Low-cost detection, Navigation path
14	RFID Reader		RC522 RFID	Scan RFID Tag	Location detection

Figure 10: Sensors

D. power System				
Sr. No.	Component	Photo	Specification	Reason for Selection
15	Battery		12V, 8Ah Lead Acid	Portable Power
16	Voltage Regulator		5V Buck Converter	Stable supply for Raspberry Pi and Arduino
18	Power Module (Latching Switch, Fuse)		Switch Protection	Safety
19	Wires (Jumper, 18 AWG and 22 AWG Copper)		30 cm Jumper wire, 18 AWG and 22 AWG Copper	Jumper for signal and copper wire for high current

Figure 11: Power System

E. Software Stack				
Sr. No.	Component	Photo	Technology	Reason
19	OS		Raspberry Pi OS 64 Bit	Stable Platform
20	Programming		Python	Easy Flexible
21	GUI		PyQt5	Touch interface
22	Vision	OpenCV	OpenCV	Image processing
23	Database	SQLite Database	SQLite	Lightweight storage
24	3D Modeling	SW	SolidWorks and Slicer	3D Object Print
25	Motor Driver		BTS7960	High Current Capacity is required to drive the heavy duty motors of the service robot

Figure 12: Software Stack

VI. IMPLEMENTATION

A. GUI Performance

The PyQt5 interface-maintained response times consistently below 2 seconds across all tested interactions. High usability was confirmed; zero training was required for first-time users, consistent with HRI design benchmarks [1].

B. Face Recognition

Recognition accuracy ranged from 90–95% under typical indoor lighting. Processing time per frame was approximately 1–1.5 seconds using the dlib 128-D pipeline. Performance degraded under extreme lighting variation consistent with findings reported in prior embedded face recognition studies [3].

C. Navigation

Throughout the test, the line following kept the lateral deviation below ±2 cm. RFID-based decision making at positional checkpoints was correct [4]. In more than 90% of cases, the scan-and-resume fault-tolerance procedure recovered effectively when the line was momentarily lost (due to foot traffic or shifted markers). The BTS7960 H-bridge driver was able to reach a travel speed of roughly 15 cm/s at 220 RPM. A signal-to-noise ratio greater than 17.30 is necessary for path tracking dependability; at this level, straight-line accuracy surpasses 98.64%.

The HARI robot achieves dependable indoor autonomous navigation by combining RFID technology, ultrasonic detection, and infrared sensors. While RFID checkpoints are utilized for location identification and destination-based decision making, the navigation system follows a predetermined black-line path placed on the floor.

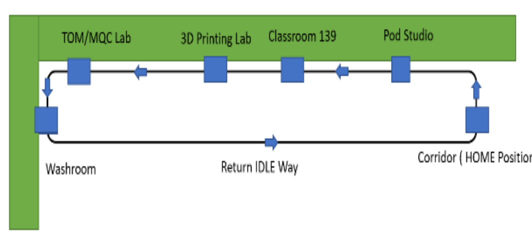


Figure13: Map Layout

By differentiating between black and white objects, infrared sensors installed beneath the robot continuously identify the travel path. The Arduino

Mega 2560 processes sensor values and uses PID-based correction and line-following logic to control motor movement to maintain stable tracking [5]. The left and right motors' speeds are independently adjusted to compensate for minor deviations from the course.

Important checkpoints including labs, classrooms, hallways, and return sites have RFID tags placed. Every RFID tag has a distinct UID that is associated with a location in the navigation database. The Raspberry Pi uses UART serial communication [6] to deliver an appropriate navigation command to the Arduino when a user chooses a location from the touchscreen GUI. The RFID scanner keeps looking for tags in the vicinity while the robot moves. The robot executes predetermined navigation action, such as halting, turning, or verifying destination arrival, as soon as it detects the matching RFID checkpoint. Include an RFID logic image.

Table 1: RFID Logic

GUI Command	Destination	RFID UID	Arduino Command
LAB1	Pod Studio	06 CE 00 04	LAB1
LAB2	Classroom 139	1E 66 00 04	LAB2
LAB3	3D Lab	04 36 01 04	LAB3
LAB4	TOM/MQC	63 34 01 04	LAB4
LAB5	Washroom	42 8C 0B 06	LAB5
HOME	Corridor	B3 2E 22 56	HOME

To identify obstacles and avoid collisions while moving, an ultrasonic sensor is incorporated. The robot's distance from surrounding objects is continuously measured by the sensor. The robot stops momentarily and waits for the road to clear before continuing to navigate if it detects an obstruction within the safety threshold. In dynamic indoor situations with human movement, this enhances operational safety.

Without the need for computationally costly SLAM or LiDAR-based systems, the navigation architecture provides accurate and stable movement by combining low-cost sensing approaches with real-time control logic. Because of this method, the HARI robot can be used in educational settings where reliability, cost, and simplicity are crucial design criteria.

D. UART Communication and Navigation Command Execution

The primary Python-based control program that generates navigation commands and facilitates GUI interaction is run by the Raspberry Pi. The Arduino Mega 2560 receives the corresponding command via UART serial connection when a destination is chosen via the touchscreen interface [2]. Successful command execution, including destination selection, navigation initiation, and serial data transfer between both controllers, is demonstrated by the terminal output displayed in the figure. Raspberry Pi's high-level decision-making and the Arduino system's low-level motor and sensor control can be reliably coordinated thanks to this communication architecture.[6]

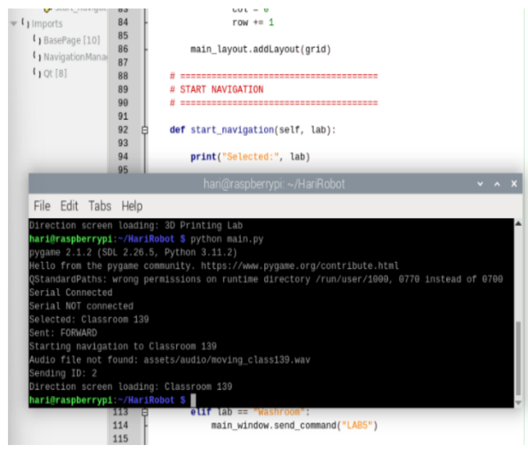


Figure 14: USB Serial UART

VII. CALCULATION & ASSUMPTIONS

1) Assumptions:

Total Weight (Robot + Load): 25kg

Number of drive wheels: 4

Wheel diameter: 75.12mm

Desired robot speed: ~3km/h

height X width: 1100 X 404 mm

Rolling resistance force (F): 14.715N

C.G:9.81m/s^2

Motor RPM: ~ 300 RPM

2) Required torque:

T= F / 4wheel x r

(14.715/4) X 0.038 = 0.1398 N/mm

Torque required per motor: ~0.140 N.m

[2],[5].

VIII. RESULT

Table 2: Comparison of HARI & Other Systems

Parameter	Existing Systems	HARI
Cost	High	Low
Navigation	Yes	Yes
Face Recognition	Limited	Limited
Interactive GUI	Partial	Full
Offline Operation	No	Yes

The Table that follows compares traditional "Existing system" with a recently created robot called 'HARI'. Cost, Navigation, Face Recognition, Interactive, GUI, and offline operation are the five main criteria by which it assesses their performance. According to the Statistics, HARI dramatically lowers costs from high to low, even though both systems offer navigation and have limited face recognition. Furthermore, HARI performs better than current systems by providing a complete interactive graphical user interface (GUI) rather than a partial one and by adding the vital feature of full offline operation.

Figure 15: Actual Deployment in Department

IX. CONCLUSION

The HARI system demonstrates that a budget-friendly, multi-functional robotic platform can be successfully built using commercially available hardware and open-source software. The dual-controller architecture cleanly separates high-level AI processing from time-critical real-time hardware tasks, with USB Serial communication providing reliable and debug gable inter-controller exchange. A lightweight fiber composite chassis reduces motor torque demand and extends battery life without compromising structural integrity.

Testing across all modules confirms stable and reliable performance: face recognition accuracy of 90–95%, GUI response below 2 seconds, and delivery task-completion of 95% over 40 runs, and lateral navigation deviation below ±2 cm. This project demonstrates that schools and colleges can adopt versatile service robots without massive budgets or infrastructure changes.

Future research will concentrate on expanding the face recognition database, integrating Wi-Fi and IoT for remote monitoring and delivery scheduling, and deploying in other institutional contexts, such as medical and logistics settings use cases that have been proven in relevant literature.

REFERENCES

[1] Mohan, "Line-following robot for automatic delivery in classrooms or libraries," *Universal Research Reports*, vol. 12, no. 3, Jul.–Sep. 2025.

[2] S.-E. Oltean, "Mobile robot platform with Arduino Uno and Raspberry Pi for autonomous navigation," *Procedia Manufacturing*, Elsevier, 2018.

[3] Aboluhom and I. Kandilli, "Face recognition using deep learning on Raspberry Pi," *The Computer Journal*, Oxford University Press, 2024.

[4] O. R. Solavat, "Path following robot using Arduino and IR sensors," *International Research Journal of Engineering and Technology (IRJET)*, vol. 11, no. 5, May 2024.

[5] S. Tutar, T. Okuyucu, and Y. Sur, "Design and implementation of a PID-based line follower robot controller," 2024.

[6] M. Z. Azeez and N. K. A. Al-Sahib, "Build and interface internet mobile robot using Raspberry Pi and Arduino," *Innovative Systems Design and Engineering*, vol. 6, no. 1, 2015.