FOSSEE Winter Internship Report

On

Industrial Robotics and Automation using Dobot

Submitted by

Anuj Naik
B.Tech Student
Fr. Conceicao Rodrigues College of Engineering

Under the Guidance of

Prof. Jayendran Venkateswaran
Department of IEOR
Indian Institute of Technology Bombay

December 2025 – January 2026

Acknowledgment

I would like to express my sincere gratitude to everyone who supported and guided me throughout the course of my internship and the successful completion of this project. Their encouragement and guidance were instrumental in shaping my learning experience.

I express my sincere thanks to Prof. Jayendran Venkateswaran, IEOR Department, IIT Bombay, for overseeing the technical aspects of the work and for his continuous guidance and encouragement.

I am also grateful to the mentors from the department, including the M.Tech students, for their valuable technical support and constructive feedback throughout the project.

I extend my gratitude to Prof. Kannan M. Moudgalya, Principal Investigator of the FOSSEE project, Department of Chemical Engineering, IIT Bombay, for enabling this internship opportunity.

I am especially thankful to my FOSSEE guides, Mr. Sumanto Kar and Mr. Varad Patil, for coordinating all official and administrative aspects of the internship and for ensuring smooth communication during its course.

I also thank the FOSSEE Managers Ms. Usha Viswanathan and Ms. Vineeta Parmar and their entire team for their consistent support.

I gratefully acknowledge the support of the National Mission on Education through Information and Communication Technology (ICT), Ministry of Education (MoE), Government of India, for facilitating this project.

Finally, I express my sincere gratitude to my college, Fr. Conceicao Rodrigues College of Engineering, and the Principal for their continued support throughout my studies.

Contents

Chapter 1

Introduction

1.1 National Mission on Education through ICT

The National Mission on Education through ICT (NMEICT) is a scheme under the Department of Higher Education, Ministry of Education, Government of India. It aims to leverage the potential of ICT to enhance teaching and learning in Higher Education Institutions in an anytime– anywhere mode. The mission aligns with the three cardinal principles of the Education Policy: access, equity, and quality.

NMEICT seeks to bridge the digital divide by empowering learners and teachers in urban and rural areas, fostering inclusivity in the knowledge economy.

1.1.1 ICT Initiatives of MoE

The Ministry of Education (MoE) has launched several ICT initiatives aimed at students, researchers, and institutions.

Table 1.1: Summary of ICT Initiatives by the Ministry of Education

No.	Resource	For Students / Researchers	For Institutions
1	SWAYAM	Earn credit via online courses	Develop and host courses
2	National Digital Library	Access e-content	Form NDL Clubs
3	e-PG Pathshala	Access free books	Host e-books
4	Shodhganga	Access research theses	List institutional theses
5	e-Yantra	Embedded systems training	Create e-Yantra labs
6	FOSSEE	Volunteer for open-source	Run FLOSS labs
7	Virtual Labs	Perform online experiments	Develop experiments
8	SAMARTH ERP	Manage lifecycle digitally	Enable e-governance

1.2 FOSSEE Project

The FOSSEE project promotes the use of Free/Libre and Open Source Software (FLOSS) tools in academia and research.

1.3 eSim Software

eSim is an open-source Electronic Design Automation (EDA) tool built using KiCad, Ngspice, and Python for circuit design and simulation.

Chapter 2

Dobot Exploration and Basic Operations

Dobot Studio Main Interface

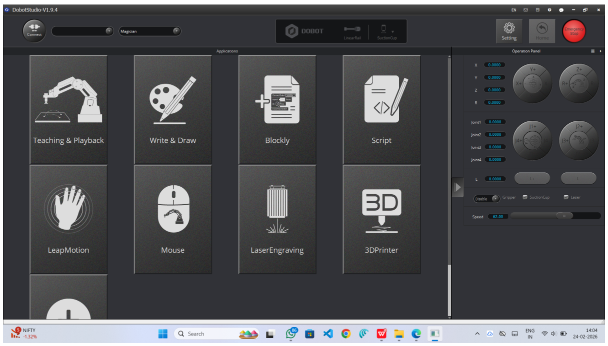

Figure 2.1: Dobot Studio

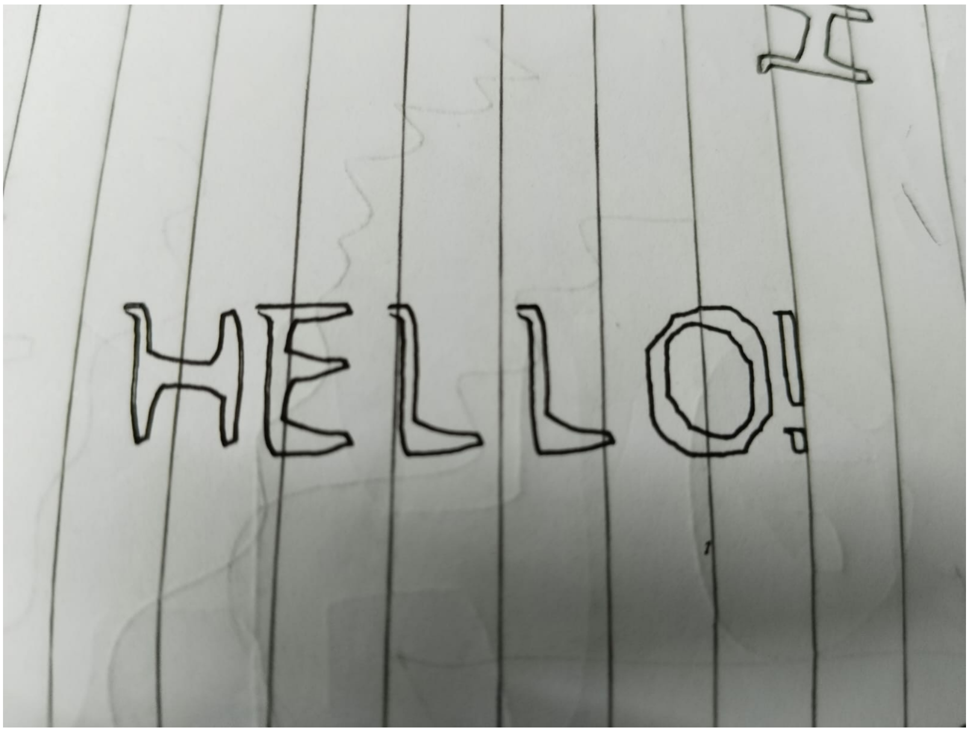

Figure 2.2: Write Draw

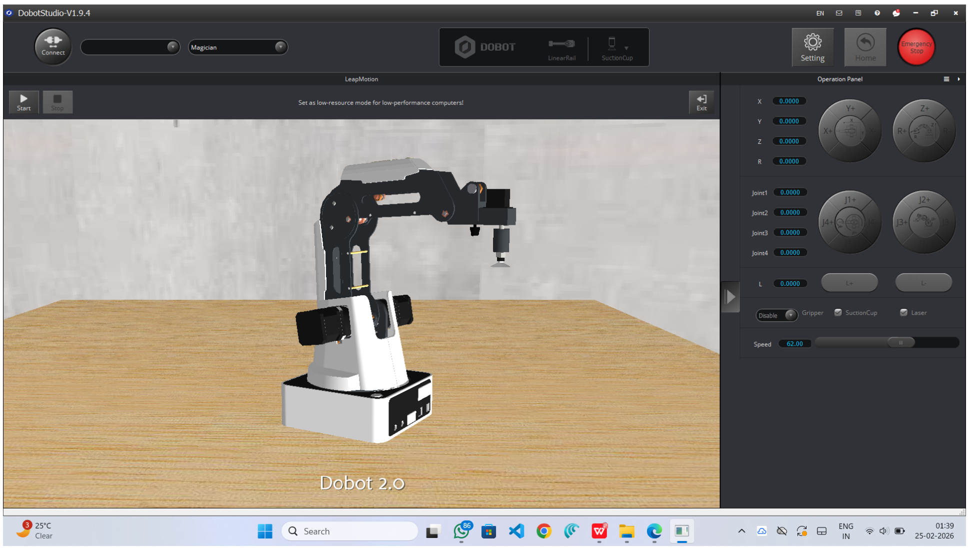

Figure 2.3: Dobot Model

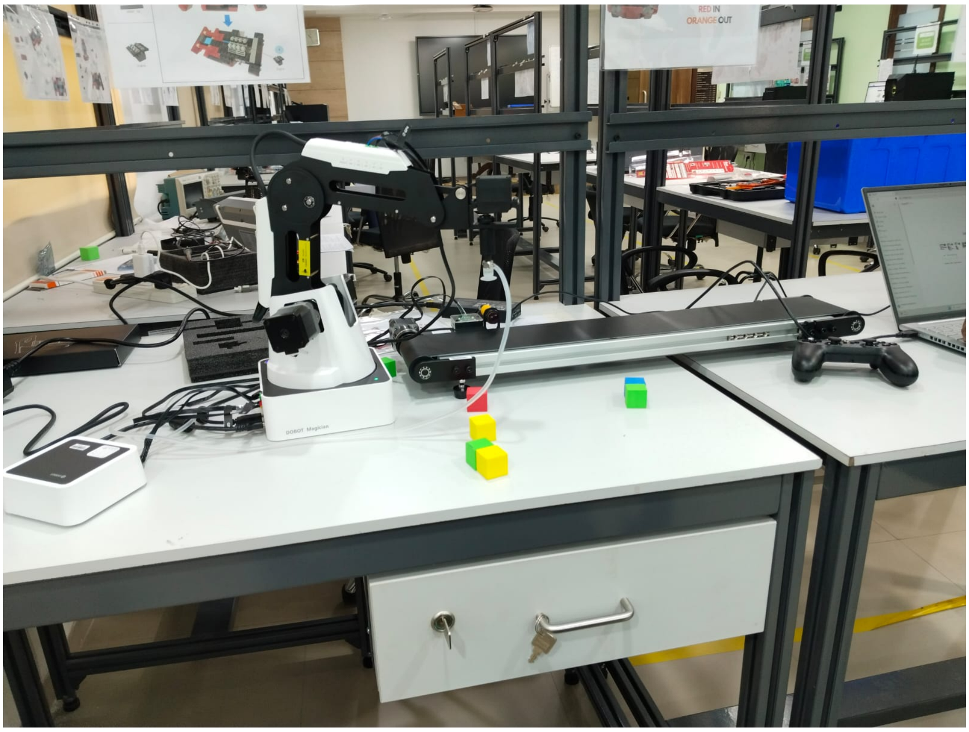

Figure 2.4: Manual Operational Control Panel

During the initial phase of the internship, detailed exploration of the Dobot Studio software environment was carried out. The objective was to understand the architecture of different operational modes, evaluate their industrial relevance, and analyze the flexibility of the Dobot robotic arm in various automation scenarios. All major control modes, except 3D Printing and Laser Engraving, were explored and implemented through practical experimentation.

2.1 Teaching and Playback Mode

The Teaching and Playback mode was used to manually guide the robotic arm to predefined coordinates and record waypoint sequences. Multiple positions were stored in memory to execute repetitive industrial-style operations such as pick-and-place and stacking.

Playback functionality ensured consistent execution of recorded motion, allowing evaluation of repeatability, smooth interpolation, and positional accuracy within the workspace.

2.2 Write and Draw Mode

The Write and Draw mode enabled the robot to perform drawing and text-writing tasks using a pen attachment. Geometric shapes such as lines, squares, and circles were executed to analyze trajectory planning and curvature control.

Text-writing operations were also performed to study stroke-based motion execution and fine calibration of tool offsets.

2.3 Blockly Programming Mode

Blockly mode provides a graphical programming interface using drag-and- drop blocks. Automation logic was created using movement blocks, delays, loop structures, and conditional statements.

This mode was used to implement structured task sequences and helped in understanding high-level automation logic without textual coding.

2.4 Script Mode

Script mode allows command-based textual programming for precise control of speed, acceleration, and coordinate movement.

Custom commands were written to test motion behavior, execution timing, and coordinate precision, providing deeper insight into Dobot's control instruction structure.

2.5 Leap Motion Control

Leap Motion mode enables gesture-based robotic control by mapping hand movements to robotic motion in real time.

This mode demonstrated intuitive human–machine interaction and highlighted latency and synchronization considerations in interactive robotics.

2.6 Mouse Control Mode

Mouse control allows the robotic arm to be manipulated using cursor movement on the interface.

Cursor displacement was mapped to Cartesian robotic motion, enabling manual precision control and better understanding of workspace mapping between digital and physical coordinates.

2.7 Operational Panel Testing

The operational panel was used extensively to test joint-space (J1–J4) and Cartesian-space (X, Y, Z, R) control.

Speed adjustment, gripper activation, and suction cup operations were analyzed to understand hardware response, safe limits, and motion scaling effects.

2.8 Summary

The exploration phase provided comprehensive understanding of Dobot's software ecosystem, including graphical programming, script-based automation, manual control, and gesture-based interaction. These foundational experiments prepared the groundwork for advanced automation and external control implementation in subsequent phases.

Chapter 3

External Control Using Arduino and Python

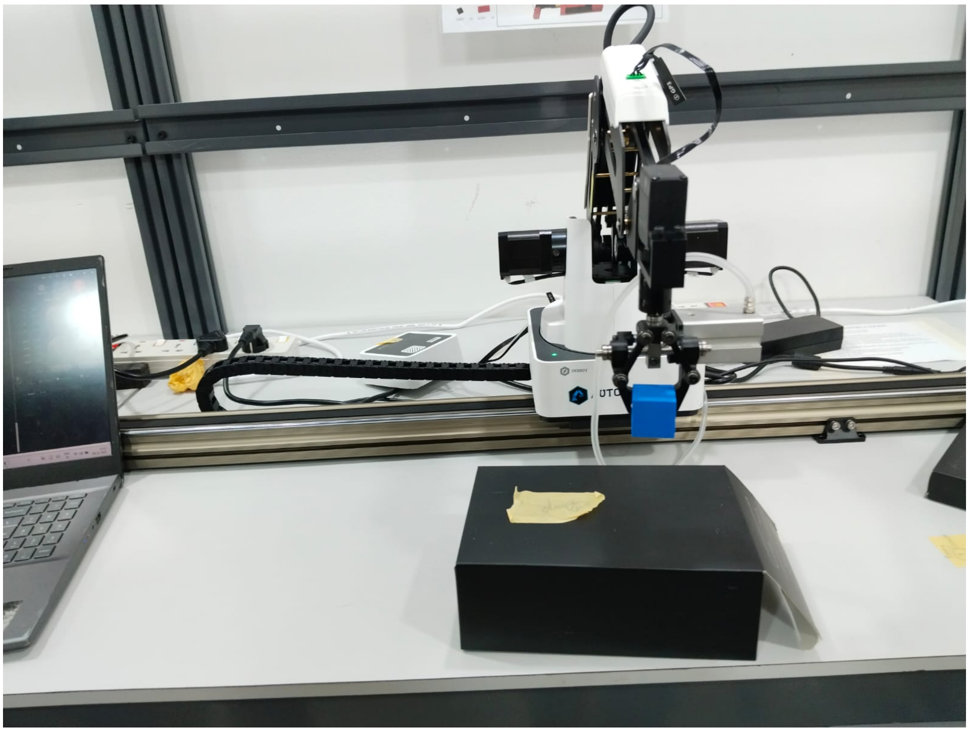

Figure 3.1: System Architecture: Python–Arduino–Dobot Communication Flow

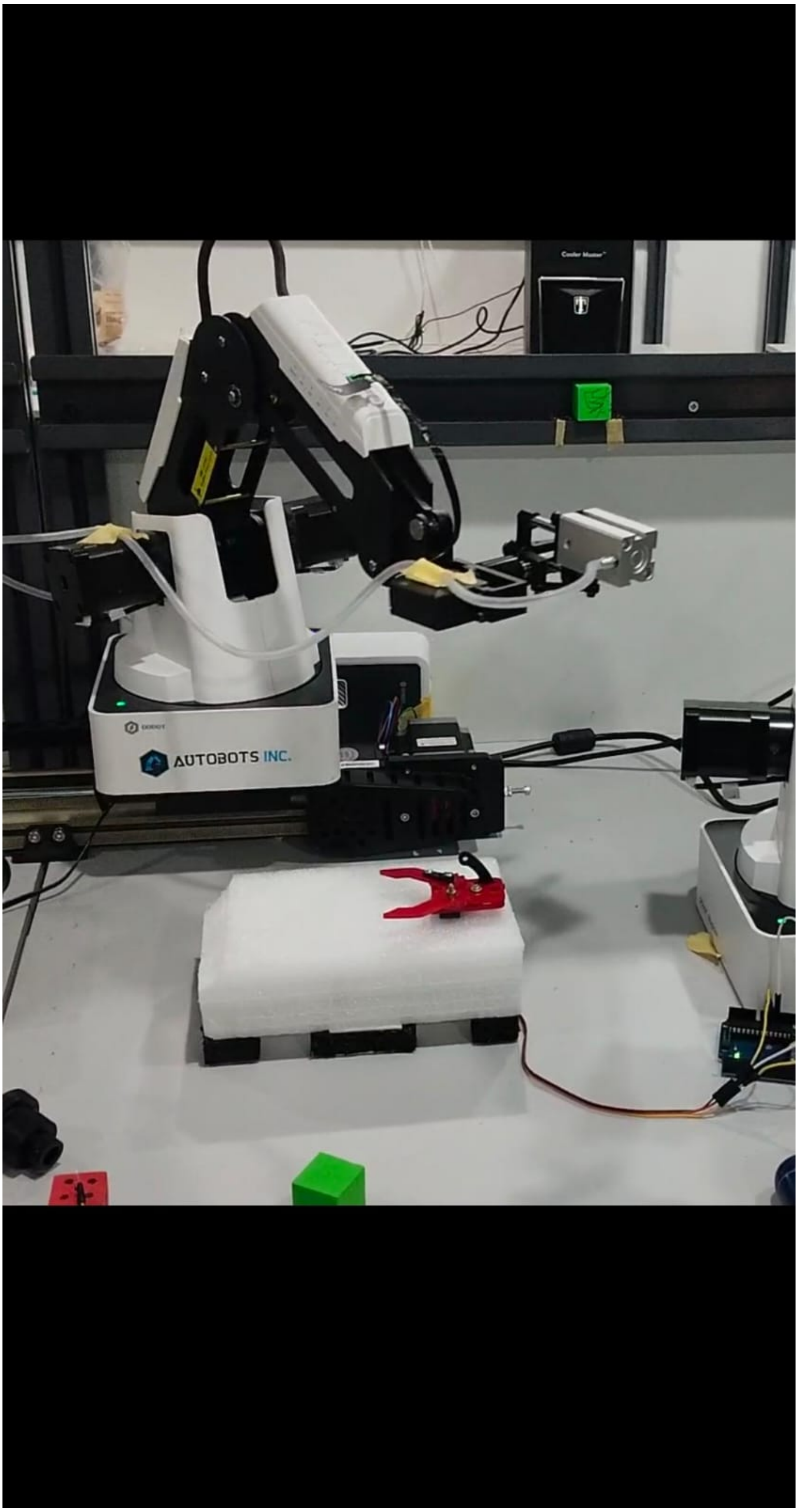

Figure 3.2: External Control Setup with Color Sensor Integration

After exploring the native Dobot Studio software environment, the next phase involved operating the robotic arm without relying on its inbuilt graphical interface. The objective was to simulate an industrial control architecture where robotic systems are controlled through external programs and embedded controllers.

3.1 System Architecture

The external control architecture consisted of three primary layers:

- Python program running on a computer
- Arduino acting as an intermediary hardware interface
- Dobot robotic arm receiving interpreted commands

A Python interpreter-based communication setup was implemented. The Arduino was programmed to receive serial data from the Python program and interpret control instructions. The interpreted commands were then forwarded to the Dobot robotic arm for execution.

3.2 Python–Arduino Communication

The Arduino was uploaded with a communication handling program that enabled it to read serial data sent from the computer. The Python script transmitted motion commands and logical instructions through the serial port.

Upon receiving the data, the Arduino parsed the instructions and executed corresponding movement commands on the robotic arm. This setup eliminated dependence on the Dobot Studio interface and allowed custom automation logic to be implemented externally.

3.3 Use of PyDobot Libraries

To simplify robotic communication and motion control, the `pydobot` and `pydobot-plus` libraries were utilized in the Python environment.

These libraries provided high-level functions to:

- Connect to the Dobot via serial communication

- Control Cartesian coordinates (X, Y, Z, R)
- Activate and deactivate end effectors
- Execute precise motion sequences

Using these libraries significantly reduced low-level communication complexity and allowed focus on automation logic.

3.4 Implemented Automation Tasks

Using the external control setup, multiple industrial-style tasks were implemented.

3.4.1 Pick and Place Operation

Python scripts were written to define coordinate-based pick-and-place movements. The robotic arm was instructed to move to a pickup location, activate the gripper or suction cup, lift the object, and place it at a target destination.

This validated the accuracy of externally controlled motion execution.

3.4.2 Stacking Operation

Stacking operations were implemented by dynamically modifying the Z-coordinate after each placement. The robotic arm successfully placed multiple objects vertically with controlled height increments.

This demonstrated precise positional control using externally generated logic.

3.4.3 Color-Based Sorting

A color sensor interfaced with the Arduino was used to detect object color. Based on the detected color value, Python logic determined the appropriate destination coordinate.

The robotic arm then transported the object to its corresponding location, effectively implementing a color-based sorting mechanism.

3.5 Significance of External Control

This phase of the internship demonstrated how robotic systems can be integrated into broader automation architectures. By separating control logic (Python), hardware interfacing (Arduino), and actuation (Dobot), a modular industrial control framework was achieved.

The implementation provided deeper understanding of serial communication, embedded system integration, and software–hardware co-design principles.

Chapter 4

Factory Setup and Industry 4.0 Concept

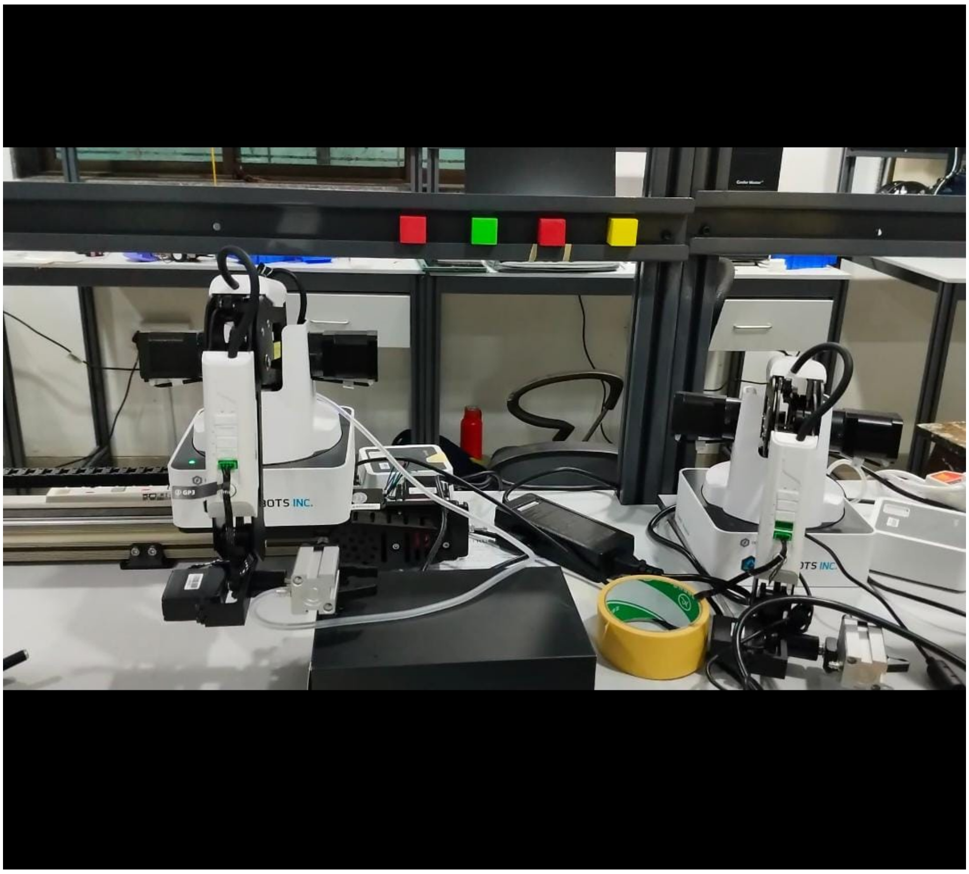

Figure 4.1: Miniature Warehouse and Factory Layout Model

Figure 4.2: Sensor and Conveyor Integration in Factory Setup

After understanding both software-based and external hardware control of the Dobot robotic arm, a miniature factory environment was designed to simulate a real-world industrial automation setup.

4.1 Warehouse Layout Design

A small-scale warehouse model was constructed using cardboard and 3D-printed components. The objective was to create designated storage locations, pickup zones, and drop-off points resembling a structured industrial environment.

The layout included clearly defined positions for raw material storage, processing area, and dispatch section. This physical segmentation helped simulate structured material flow within a production system.

4.2 Conveyor and Material Flow

A conveyor mechanism was integrated into the setup to mimic automated material transportation. The robotic arm was positioned strategically to interact with objects arriving via the conveyor.

This allowed simulation of sequential operations where materials move from storage to processing and finally toward dispatch. The conveyor system enhanced realism and demonstrated continuous production flow.

4.3 Sensor Integration

Sensors were integrated into the setup to enable intelligent decision- making within the system. A color sensor was used to detect object attributes and determine sorting destinations.

The sensor data was processed through Arduino, and corresponding control logic in Python directed the robotic arm accordingly. This demonstrated real-time data acquisition and control response.

4.4 Industry 4.0 Concept Implementation

The overall setup was designed to align with Industry 4.0 principles, which emphasize smart manufacturing, automation, and interconnected systems.

Key Industry 4.0 elements demonstrated in this setup include:

- Integration of physical and digital systems
- Real-time sensor-based decision making
- Modular automation architecture
- Data-driven control logic

By combining robotic automation, sensor feedback, and external programming, the miniature factory model represented a simplified cyber-physical production system.

4.5 Significance of the Factory Model

The factory setup transformed isolated robotic tasks into a coordinated automation workflow. It provided insight into production line organization, material handling strategies, and structured automation architecture.

This phase bridged the gap between standalone robotic experimentation and practical industrial system design.

Chapter 5

Final Project: Customized Keychain Manufacturing System

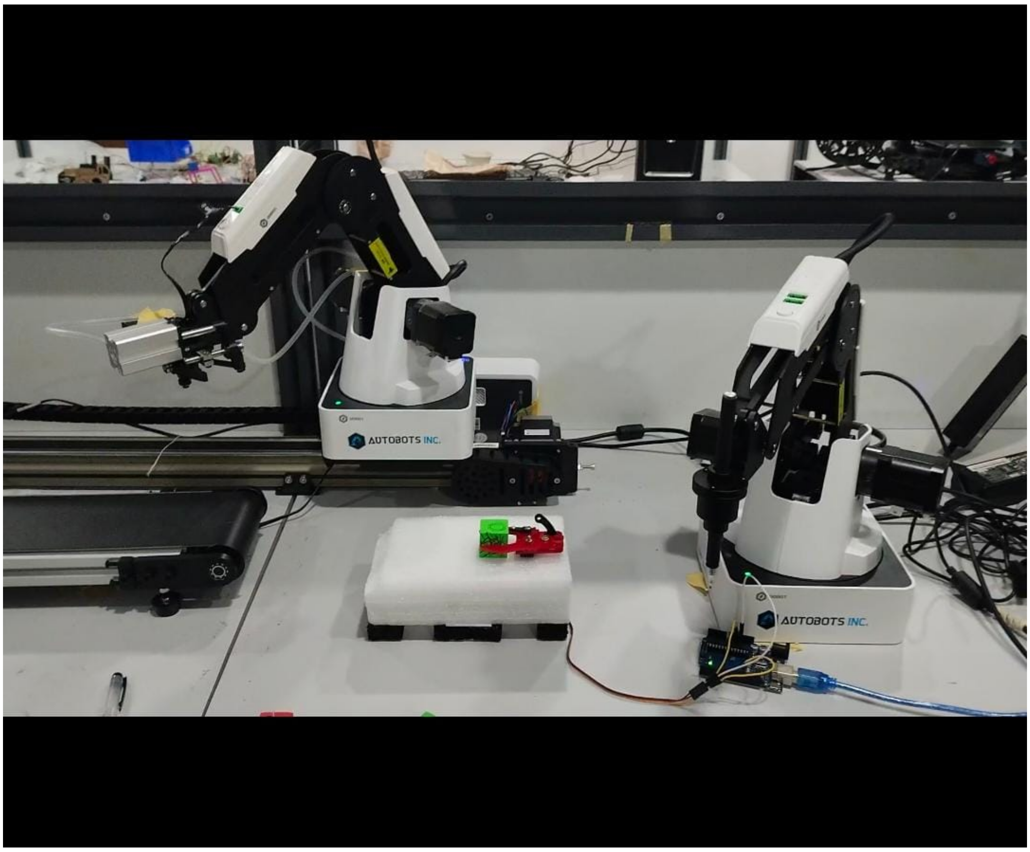

Figure 5.1: Overall Workflow of Customized Keychain Manufacturing System

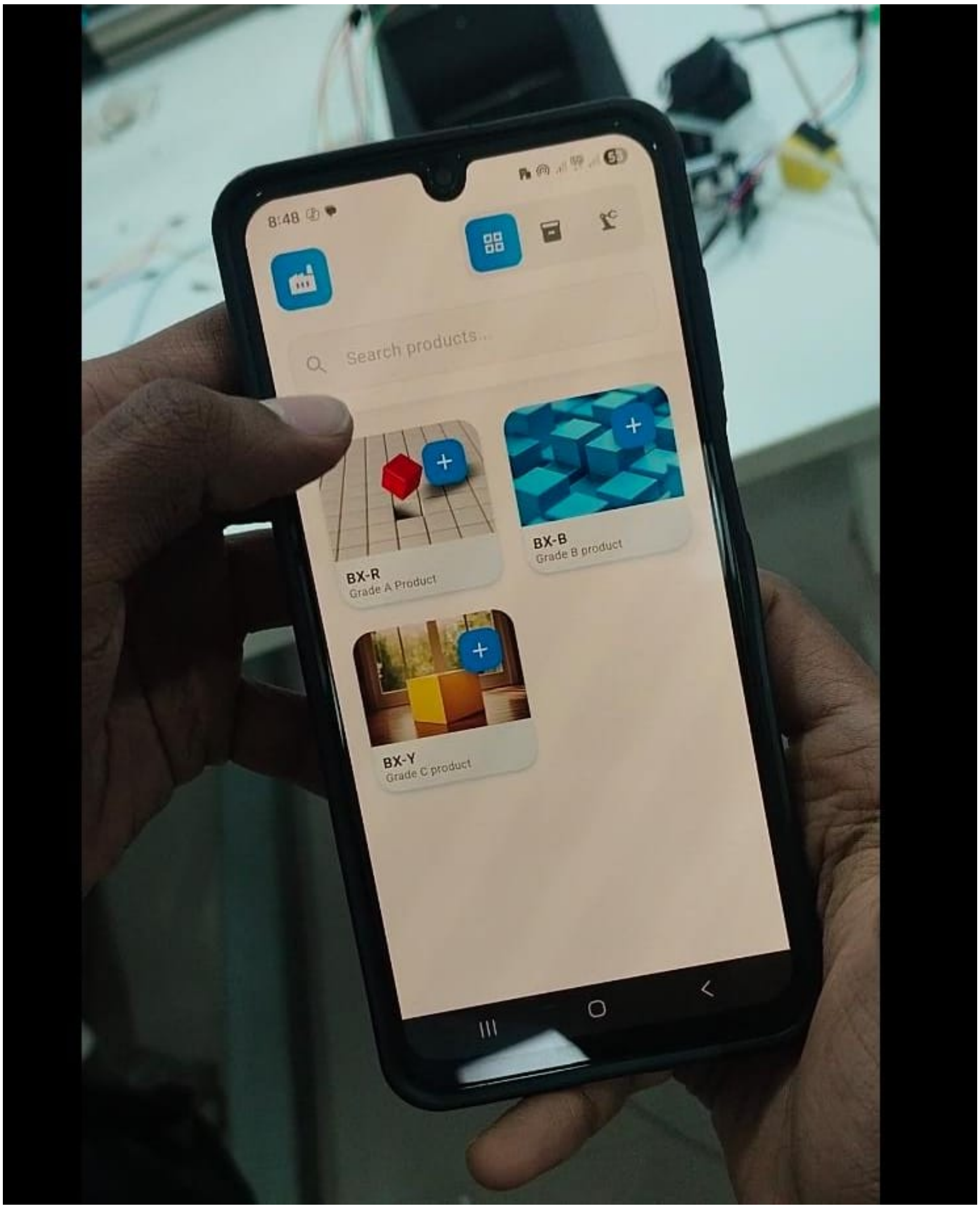

Figure 5.2: App Interface

The final project of the internship involved the design and implementation of a miniature automated manufacturing system for customized keychains. The objective was to simulate a smart production workflow where user input directly influences manufacturing operations in real time.

5.1 Problem Statement

The goal was to design a system where a user could select a keychain color and customized text through an application interface. Based on this input, the system would automatically retrieve the selected material, process it, and deliver the finished product to the dispatch section without manual intervention.

5.2 Application-Based Control Interface

A simple application interface was developed to allow users to:

- Select the color of the keychain
- Enter customized text for engraving/writing

The application transmitted user input to the control system, which processed the data and generated corresponding robotic instructions. This demonstrated integration between a digital user interface and physical automation hardware.

5.3 Two-Dobot Coordination

The manufacturing system incorporated two Dobot robotic arms working in coordination.

Dobot 1 – Material Handling Unit

The first Dobot was responsible for:

- Picking the selected colored keychain block from warehouse storage
- Placing it onto the processing station

It operated based on the color selected in the application interface. The appropriate storage coordinates were selected dynamically using predefined mapping logic.

Dobot 2 – Writing and Processing Unit

The second Dobot was dedicated to writing customized text on the keychain using the Write and Draw mode.

Based on the user-provided input, motion paths were generated to write the specified text accurately on the keychain surface. This demonstrated precise trajectory planning and coordinated task segmentation between two robotic systems.

5.4 Conveyor Integration

After processing, the completed keychain was placed onto a conveyor system. The conveyor transported the product toward the dispatch area, simulating automated product movement within a production line.

This eliminated manual transfer between stations and created a continuous material flow within the system.

5.5 Forklift (AGV) Control

To enhance industrial realism, a miniature forklift/AGV (Automated Guided Vehicle) mechanism was integrated into the setup.

The forklift unit was programmed to:

- Receive the processed keychain from the conveyor endpoint
- Transport it to the designated warehouse or dispatch zone

This demonstrated multi-agent coordination and extended the automation chain beyond stationary robotic arms.

5.6 System Workflow

The complete workflow of the system can be summarized as follows:

1. User selects color and text in the application.
2. Control logic identifies warehouse location of selected material.
3. Dobot 1 retrieves and places the keychain at processing station.
4. Dobot 2 writes customized text.
5. Finished product is transferred to conveyor.
6. Conveyor moves product to dispatch endpoint.
7. Forklift transports final product to destination zone.

5.7 Industry 4.0 Alignment

The project embodied several Industry 4.0 principles:

- Cyber-physical system integration
- Real-time digital-to-physical execution
- Multi-robot coordination
- Modular automation architecture
- Smart material handling

By combining application control, robotic coordination, sensor logic, and automated transport, the system represented a scaled-down model of a smart manufacturing environment.

5.8 Project Significance

This project integrated all prior learning phases — software exploration, external control, sensor integration, and factory layout design — into a unified intelligent automation system.

It demonstrated practical implementation of coordinated robotics, distributed control, and real-time manufacturing logic within a miniature industrial setup.

Chapter 6

Learning Outcomes

The internship provided extensive exposure to robotics, automation, embedded systems, and smart manufacturing concepts. The following learning outcomes summarize the technical and conceptual knowledge gained during the internship period.

6.1 Understanding of Robotic Control Systems

Through exploration of Dobot Studio and external programming, a comprehensive understanding of robotic coordinate systems, joint-space versus Cartesian control, and motion planning was developed.

Hands-on experimentation with waypoint recording, script-based movement, and graphical programming strengthened the practical understanding of robotic positioning and repeatability.

6.2 Software–Hardware Integration

One of the most significant learning outcomes was the ability to integrate software logic with physical hardware components. The use of Python for control logic and Arduino for serial communication established a clear understanding of embedded system architecture.

This experience improved knowledge of communication protocols, data parsing, and modular automation design.

6.3 Industrial Automation Concepts

Designing the miniature factory setup provided insight into production line organization, material flow optimization, and automation workflow structuring.

The implementation of conveyor systems, sensor-based sorting, and robotic coordination reflected real-world industrial automation principles.

6.4 Multi-Robot Coordination

The final project involving two Dobot robotic arms enhanced understanding of task segmentation, synchronization, and distributed control.

Coordinating material handling and processing tasks between multiple robotic systems highlighted the importance of workflow planning and execution timing.

6.5 Industry 4.0 Awareness

The internship introduced core Industry 4.0 concepts such as:

- Cyber-physical systems
- Smart manufacturing
- Real-time data-driven decision making
- Modular automation architecture

The integration of application-based control with robotic execution demonstrated digital-to-physical system interaction.

6.6 Problem-Solving and System Design Skills

Throughout the internship, several technical challenges were encountered, including communication synchronization, coordinate calibration, and motion optimization.

Solving these challenges improved analytical thinking, debugging skills, and system-level design capability.

6.7 Overall Technical Growth

The internship significantly enhanced practical exposure to robotics and automation beyond theoretical coursework. It strengthened programming skills, embedded system knowledge, and industrial automation understanding.

The experience built confidence in designing and implementing complete automation systems from conceptualization to execution.

Chapter 7

Conclusion

The Winter Internship at IIT Bombay under the FOSSEE initiative provided comprehensive exposure to robotics, automation systems, and smart manufacturing concepts. The internship progressed in a structured manner, beginning with software exploration, advancing to external hardware-based control, and culminating in the development of a coordinated multi-robot manufacturing system.

The initial phase focused on understanding Dobot Studio and its various operational modes, including graphical programming, script-based control, and manual coordinate manipulation. This phase established a strong foundation in robotic motion planning and system control.

The second phase introduced external control using Python and Arduino, enabling modular automation architecture. Through the use of serial communication and PyDobot libraries, the robotic arm was successfully controlled without relying on its native software interface. This significantly enhanced understanding of embedded systems and software–hardware integration.

The development of the miniature factory setup further expanded the learning scope by incorporating conveyor systems, sensor-based decision making, and structured material flow. This phase bridged the gap between isolated robotic experiments and real-world industrial automation environments.

The final project — the Customized Keychain Manufacturing System — integrated application-based control, two-Dobot coordination, conveyor automation, and forklift transport mechanisms. The system demonstrated a scaled-down implementation of Industry 4.0 principles, including cyber-physical integration, real-time execution, and smart workflow automation.

Overall, the internship strengthened practical knowledge in robotics, automation architecture, embedded communication, and intelligent manufacturing systems. It provided valuable hands-on experience in designing, implementing, and analyzing complete automation solutions within a structured industrial framework.

The experience gained through this internship will serve as a strong foundation for future work in robotics, automation, and advanced manufacturing technologies.

The complete source code, implementation files, and documentation for the projects developed during the internship are available at:

GitHub Repository:
`https://github.com/SmartDeve/IIT-IEOR-Lab`

Demonstration videos, system walkthroughs, and testing recordings are available at:

Google Drive Folder:
`https://drive.google.com/drive/folders/1I836Ifn8df5Zflaje_ec24CngTVtIiff`

Bibliography

[1] https://www.nmeict.ac.in

[2] https://fossee.in

[3] https://esim.fossee.in

[4] https://docs.arduino.cc

[5] https://docs.kicad.org