American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-14, Issue-1, pp-01-06
www.ajer.org
Research Paper Open Access

Design and Implementation of a Raspberry Pi Based Intelligent Logistics Sorting and Storage System

Junzhe Ma, Jinghao Xu, Yuhang Jiang, Minxiu Guan
School of Intelligent Manufacturing, Huzhou College, Zhejiang, China
Corresponding author's (Junzhe Ma)

Abstract: *Given that the majority of logistics sorting systems currently lack a comprehensive lightweight model for experimental research, this project uses express sorting as an application scenario and investigates a collaborative control system comprehensively that combines QR code recognition cargo classification and robot arm control. A miniature version of an intelligent sorting device is created that can automatically recognize and classify the items, lift the robot arm, and position it in the designated spot. When the items are put on the conveyor belt, identified by the camera on the gantry, sends the information to the control terminal, where the internal classification algorithm sorts them. Once the classification is finished, the signal is sent to the mechanical arm, which picks items up and positions them in the proper spot. In addition to offering a small experimental platform for the research of this type of system, the device can largely reflect the primary function of logistics sorting and warehousing systems. This helps to increase the experimental efficiency of this type of research, which in turn supports rapid development of the logistics industry.*

Key words: *logistics sorting; Raspberry PI; Robot arm; Intelligent control*

Date of Submission: 03-01-2025 Date of acceptance: 14-01-2025

I. INTRODUCTION

Intelligent logistics has been considered as a crucial industrial sector to achieve the high levels of industrial development, which is a key component mentioned in Made in China 2025. [1, 2]The express delivery sector over the world has grown rapidly in recent years, which presents not only a significant potential but also an enormous challenge for Chinese express delivery sector. Radio Frequency Identification (RFID) technology is the foundation of many automatic sorting systems in other countries. This type of device scans the bar code on package using infrared radiation to get the information of it. The package is subsequently sent across the conveyor belt within the system to its destination through different exits. Since China started sorting and transporting goods late, research about automatic sorting devices is still in its early stages, and cargo sorting technology is still in its infancy, most express companies have not fully embraced the automatic identification and sorting system.

With the development of the times, there is a vast amount of logistics data internationally, necessitating the sorting and processing of the data at the sorting station. However, it is challenging to achieve the precise positioning of sorted parcels due to the huge size and fixed overall mode of the current sorting equipment. Nevertheless, most of the sorting equipment uses transportation sorting, which only serves the purpose of transportation, meaning that the sorting work is still in the initial stage and that manual secondary sorting is then required. Another factor is that plenty of packages contain fragile items that need to be handled with extreme caution. In other words, the transport cannot be adjusted speed, and the violent transfer process can damage the package. Therefore, the design of a set of low-cost, efficient and intelligent logistics identification and classification device can ensure that under the current situation of increasing labor costs, the automation degree of express sorting system can be improved by replacing manual labor with machines[3, 4], which will reduce the overall operating capital of most express delivery industries in China. It is of great practical significance to promote its transformation from manual labor type to technology automatic type[5].

II. MATERIALS AND METHODS

1. Hardware system design

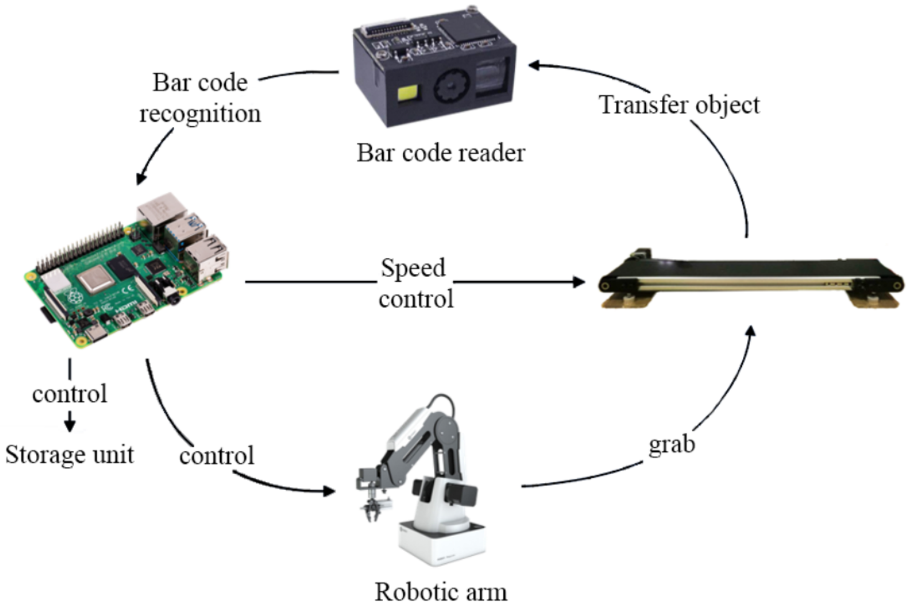

Figure 1: Hardware connection diagram

1.1 Raspberry PI module design

Raspberry PI (with the operating system installed), display (compatible with Raspberry PI), controller for the Dobot robot with mechanical arm, and attach cords (the Raspberry PI and monitor are connected via HDMI connections) were required during this project. The display screen should be connected to the designated interface of the Raspberry Pi via an HDMI cable. Furthermore, the controller of the Dobot robotic arm needs to be connected to the Raspberry Pi through the USB interface. The first step is to start the Raspberry Pi and the display screen to ensure that the display screen can normally display the output of the Raspberry Pi. Install and configure the software libraries or drivers for communicating with the robotic arm controller. Then write a program to control the robotic arm using Python. Import the necessary libraries, establish a communication connection with the robotic arm controller, define the action functions of the robotic arm, such as moving, grasping, rotating, etc. Control the actions of the robotic arm by sending instructions through the program. Logical judgments can be added to the program to perform different actions according to different situations. The status information of the robotic arm can be displayed in real time using the display screen.

2.2 Mechanical arm module design

In this project, the DOBOT Magician manipulator was selected as the core execution device of the sorting system. The robot arm was controlled by the Python SDK, which supported initialization and motion control of the robot arm, improving the flexibility of the control. A Python environment was used in the main controller to run control programs, and the robot arm was connected to the Raspberry PI using a USB interface. After the system started, the connection of the robot arm was initialized through Python SDK, and the communication that linked with the control host was established to ensure the control instructions could be transmitted to the robot arm in real time.

The desired grabbing position was pre-set by the control system, while the material moved along the conveyor belt and reaches the predetermined position, the system could activate the grasping operation of the mechanical arm. During the implementation process, the control system determined the current location of the material through the time interval based on to the speed of the conveyor belt and the initial position of the material departure. The control system instructed the robot arm to do the grasping motion when the calculation results indicate that the material has arrived at the designated gripping location. This approach efficiently streamlines the hardware configuration of the system, which relying on accurate velocity computation to finish the position detection.

Python was used to programed the movement track code of the robot arm based on the speed of the conveyor belt and the location of the material, this design included a series of operation steps including grasping, handling and placing. In specific operation, the motion trajectory of the robot arm included the following three steps:

Grasp operation: The robot arm adjusted the motion path based to the material position information provided by the control system, so that the suction tool was located in the appropriate position above the material.

Handling operation: After the grasp was completed, the mechanical arm would transport the material to the specified storage unit location in accordance with the preset path.

Placing operation: The sucking tool was controlled by the robot arm via the SDK to release the material and finish the placing operation once it had reached the target position.

2.3 Storage module design

One of the highlights in this project was the design of the storage module. In view of the problems of bulky storage equipment and poor flexibility on the market, the project team has designed a modular storage unit, which can be easily disassembled and assembled freely, improving the adaptability of the equipment in various environments. The diagram of the storage module can be seen in figure 2.

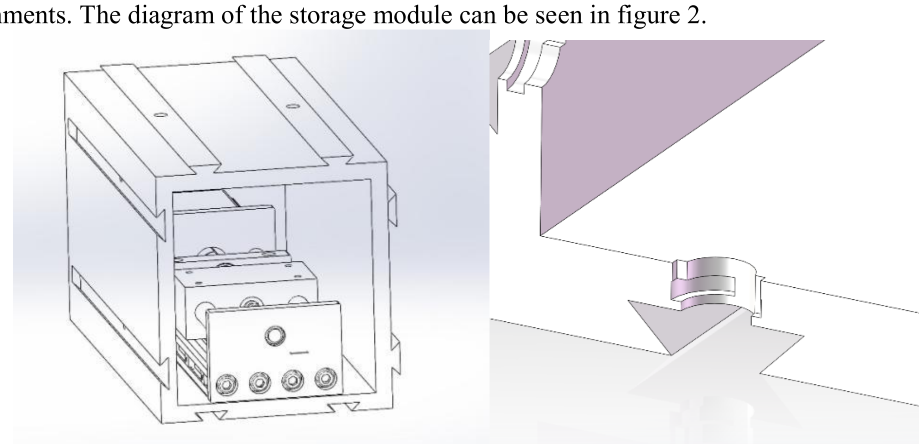

Figure 2: Design for Storage Unit

A servo motor installed on the inner bottom of the housing, a storage box mounted on the servo motor slide, and a housing make up the storage unit. The main feature of its structure was that the upper and right sides of the outer surface of the shell are designed with two parallel convex transverse slide rails, while the left side and bottom have two parallel chutes, and all these units can be chimed with each other. To give the impression that the unit was fixed when the two units are ccombined, a unique bolt hole is made in the chute and was fitted with a unique bolt that was inserted and rotated. The latch described above was distinguished by having two symmetrical protrusions on the left and right sides. These protrusions can be rotated into the latch hole after the latch was in position. Thus, flexible disassembly achieves its intended purpose.

2. System software design

2.1 Raspberry PI control board programming

In order to program and control the stepper motor, the project team used Python on the Raspberry Pi 3B to connect via USB to the MegaPi controller. The code can be found in figure 3 below.

```
from megapi import *

bot=MegaPi()
bot.start(port="/dev/ttyUSB0")
bot.stepperMotorRun(1,3000);

while 1:
    bot.stepperMotorRun(1,3000);
    sleep(1);
```

Figure 3: Control program excerpt

Although "/dev/ttyUSB0" is commonly used as the USB device name in the given code, this identification may differ in real-world usage.

2.2 Robot arm communication pairing | human-computer interaction interface design

In this project, the communication between the robotic arm and the Raspberry PI was realized through Python SDK. The specific process is as follows:First, the DLL file of Dobot was loaded into memory and the corresponding CDLL instance was obtained in order to call the manipulator's control function. Then, establishing a serial connection with the Dobot arm using the ConnectDobot function and set the baud rate to 115200 was necessary. During connection, the program returned to the connected status to determine whether the connection was occupied. Common return states included DobotConnect_NoError (indicating connection successful), DobotConnect_e (indicating device not found), and DobotConnect_Occupied(device occupied).To make sure that every operational instruction was carried out in the correct order, the system cleared the instruction queue of Dobot arm when a successful connection had been made. To guarantee the motion of robot arm was accurat enough in ensuing operations, the system would then configure the motion parameters of robot arm, such as joint parameters and return to zero position.After setting the parameters, the system sent a series of motion instructions to the robot arm, which were executed asynchronously by the SetPTPCmd function and added to the instruction queue of the Dobot. The system would continuously monitor the execution progress of the instructions until all the instructions had been executed.The system disconnected from the Dobot arm and ended the command queue of robot arm after the execution was completed. This allowed the robot arm and Raspberry PI to communicate effectively and consistently, thus achieving precise control of the robot arm to meet the needs of express sorting applications in the project.

2.3 Barcode data reading

The project team adopted MJ-2000 agile code scanning gun. After studying the transmission and the actual work flow of the code scanning gun, the working flow chart in Figure 4 was designed. Through the Raspberry PI programming, when the object was captured and recognized by the sensor, the system would determine whether the object had a readable barcode and obtain all the information contained in the barcode, which further instructed the robot arm and the storage unit to take the next action.

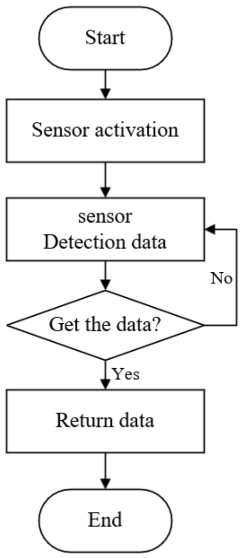

Figure 4: Process of reading barcode

2.4 System workflow design

According to the actual situation, the project team designed the work flow of the intelligent sorting system. As shown in the figure, the system obtained the detailed information of the object through the "one object, one code" by the QR code identifier in front of the conveyor belt. Under the control of the Raspberry PI, the corresponding storage unit was ejected. When the object was sent to the designated location, the mechanical arm grabed and placed it in the corresponding unit. At the same time, the background carried out statistics on the sorted goods, and displayed the statistical data on the interactive storage screen.

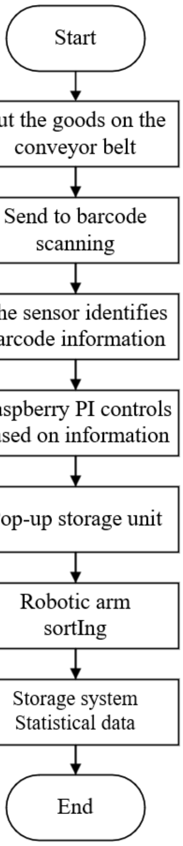

Figure 5: System flow chart

III. RESULTS AND DISCUSSION

1. Experimental test
1.1 Experimental platform construction

The system was constructed in kind by the project team. In order to make sure that every part and module of the system could fulfill the project goals, the project team first conducted thorough design planning. The prototype system was constructed at the specified location once the necessary hardware (in figure 6) and software equipments were acquired in accordance with the design scheme.

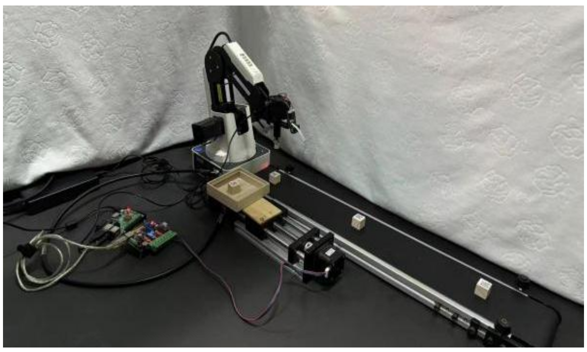

Figure 6: the hardware equipments during the project

1.2 Logistics identification sorting device operation test

The project team designed several sets of experiments to verify the accuracy and efficiency of the system, respectively changing the conveyor speed (Vc) and robotic arm activity speed (Vj), in addition to exploring the sorting situation of the two under different speed combinations. The following is the experimental table and data.

Table 1: Accuracy at different conveyor speeds

Destination	Vc1	Vc2	Vc3
A	100%	100%	100%
B	100%	100%	100%
C	100%	100%	100%

Table 2: Accuracy at different robotic arm speeds

Destination	Vj1	Vj2	Vj3
A	100%	100%	100%
B	100%	100%	100%
C	100%	100%	100%

Table 3: Accuracy at different speed combinations

	Vj1	Vj2	Vj3
Vc1	100%	100%	100%
Vc2	100%	100%	100%
Vc3	100%	100%	100%

Through the tables above, it can be seen from the experimental results of the three groups that the sorting accuracy of the system under different speed conditions met the expectation and the working condition was good. In addition, according to the test results of the third group, the system still achieved the expected sorting accuracy under different speed combinations, which can met the demands of changing the equipment speed under different sorting conditions.

IV. CONCLUSION

This project had successfully designed and implemented a small intelligent logistics sorting and storage system based on Raspberry PI, which has shown remarkable efficiency and accuracy in experimental research. Through in-depth research on the application scenarios of express sorting, a collaborative control system that integrates cargo classification with QR code recognition and robotic arm control was developed. This system not only provided a small experimental platform for the research of logistics sorting and storage system, but also reflected the main functions of logistics sorting and storage system, so as to improve the experimental efficiency of such research and support the rapid development of logistics industry.In terms of hardware system design, the precise control of the robot arm was designed through Python programming, and realized the effective interaction of the storage module. In the software system design, Python SDK was used to realize the communication between the robot arm and the Raspberry PI, and realized the automatic identification and classification of items through the two-dimensional code recognition technology. The experimental results show that the system can maintain 100% sorting accuracy at different conveyor speed and manipulator speed, which proved the stability and reliability of the system. In addition, the system could still maintain the expected sorting accuracy under different speed combinations, and could meet the needs of equipment speed changes under different sorting conditions.

In general, the intelligent logistics sorting and storage system developed in this project had important practical significance in improving sorting efficiency, reducing labor costs and enhancing logistics automation. Future work will focus on providing more solutions to further optimize system performance, expand functions and explore more practical application scenarios.

ACKNOWLEDGEMENT

Zhejiang University Student Science and Technology Innovation Activity Program (New Talent Development Initiative) (NO.2023R463001)

REFERENCE

[1]. Liu, W., et al., Evaluation of the intelligent logistics eco-index: evidence from China. 2020. **274**: p. 123127.
[2]. Sharma, D. and K. Tripathi, Navigating the Digital Age: Smart Manufacturing and Logistics, in Recent Trends in Artificial Intelligence Towards a Smart World: Applications in Industries and Sectors. 2024, Springer. p. 99-144.
[3]. Liu, Y., et al., Artificial intelligence in smart logistics cyber-physical systems: State-of-the-arts and potential applications. 2023. **1**: p. 1-20.
[4]. Zhai, C., et al. The Application of Intelligent Sorting Equipment in Warehouse Logistics. in 2023 2nd International Conference on Mechanical Engineering and Power Engineering (MEPE). 2023. IEEE.
[5]. Chung, S.-H.J.T.R.P.E.L. and T. Review, Applications of smart technologies in logistics and transport: A review. 2021. **153**: p. 102455.