FILE SORTING SYSTEM WITH IOT

A Report on Project Phase II

ABHAY C (CHN19AE001)

in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in Electroics and instrumentation Engineering

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Department of Electronics Engineering

College of Engineering, Chengannur

Chengannur - 689121, India

May 2023

DEPARTMENT OF ELECTRONICS ENGINEERING
COLLEGE OF ENGINEERING, CHENGANNUR
ALAPPUZHA - 689121

CERTIFICATE

This is to certify that this report entitled ***' FILE SORTING SYSTEM WITH IOT'*** *is a bonafide record of the study carried out by* ***ABHAY C****, under my supervision in the Department of Electronics Engineering, College of Engineering, Chengannur. The results presented in this report or parts of it have not been presented for the award of any other degree.*

Mr. VIBESH V PANICKER	**Mr. YASIM KHAN M**	**Dr. LAILA D**
Assistant Professor	**Assistant Professor**	**Head of the Department**
Project Guide	**Project Supervisor**	**Electronics Engineering**

Chengannur

Date: 10/05/2023

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to **Dr.Smitha Dharan**, Principal, **Dr Laila D**, Head of the Department, Electronics Engineering and our staff advisor **Mr. Yasim Khan M** for granting his kind consent for carrying out the suggested project. Also I would like to express my gratitude to my project coordinators **Mr. Yasim Khan M**, Assistant Professor, Department of Electronics Engineering for the incites and encouragement to improve the project. I thank our project guide **Mr. Vibesh V Panicker**, Assistant Professor, Department of Electronics Engineering for all his support.

I am greatly beholden to all the faculty members of the Department of Electronics Engineering for their wonderful assistance.Finally, I am grateful to my parents, who gave me tremendous support and forbearance throughout my work. I specially thank all my dear friends, for their help and encouragement throughout the work.

ABSTRACT

Efficient file management is crucial in office and administrative sectors to ensure productivity and streamlined operations. This project presents an innovative File Sorting Machine that leverages Internet of Things (IoT) technology to automate the process of categorizing files based on the dominant color of their cover page. By utilizing a Raspberry Pi and a camera module, the system captures images of incoming files, applies advanced image processing algorithms to analyze the dominant colors (R, G, B) present on the cover page, and accurately categorizes the files into collectors designated for finance, tax, and education.

The File Sorting Machine significantly improves efficiency and accuracy in file management by automating the sorting process in office and administrative sectors. This is achieved through the utilization of IoT technology and advanced image processing, resulting in enhanced productivity, streamlined operations, and seamless coordination among employees.

Furthermore, the File Sorting Machine extends its applicability beyond office and administrative sectors to libraries, educational institutions, legal offices, and various industries where efficient file organization is paramount. The technology provides time savings, improved organization, and enhanced accessibility to relevant files. Its scalability and adaptability allow for customized file categorization based on specific requirements.

Overall, the IoT-based File Sorting Machine revolutionizes file management, offering a reliable solution for efficient organization in diverse professional settings.

Contents

List of Figures

List of Tables

1. INTRODUCTION

This project introduces an innovative File Sorting Machine designed for efficient file organization in office and administrative sectors. The machine leverages IoT technology and image processing techniques to automate the categorization of files based on the dominant color of their cover page.

The File Sorting Machine features a conveyor belt system where users or applicants can place their files one at a time. Positioned above the belt, a camera captures images of the cover page, enabling the system to analyze the dominant colors (R, G, B) present.Based on the color analysis results, a servo motor is activated to divert the file to the corresponding compartment or section designated for its type.

The system integrates an IoT platform to enable real-time monitoring and reporting. A Raspberry Pi serves as the central hub, establishing communication between the File Sorting Machine and the IoT platform. This allows the machine to upload data regarding the count and categorization of received files, which can be accessed by employees through a user-friendly web-based interface or dedicated application.

1.1 Problem statement

The conventional method of file organization in office and administrative sectors is plagued by several drawbacks. The manual sorting process is time-consuming, re-

quiring employees to handle each file individually, review its contents, and determine the appropriate category or section for placement. This approach is prone to human errors, such as misplacement or misfiling of documents, leading to difficulties in locating specific files when needed. Additionally, manual sorting lacks efficiency, often resulting in delays in accessing required documents and decreased overall productivity. Inconsistent categorization practices among employees further complicate the process, causing confusion and hindering file retrieval. Scaling the conventional system to accommodate a larger volume of files becomes challenging, requiring additional resources and manpower. Moreover, the lack of real-time information on the count and categorization of received files hampers decision-making and overall efficiency. Accessibility and retrieval issues also arise, as locating specific files within the manual filing system can be time-consuming and error-prone.

1.2 Objectives

Automation: The primary objective is to automate the file sorting process in office and administrative sectors, reducing the reliance on manual labor and enhancing operational efficiency.

Efficient Organization: The application aims to improve file organization by accurately categorizing files based on the dominant color of their cover page. This ensures that files are stored in the appropriate compartments or sections, enabling easy retrieval and minimizing errors.

Time Savings: By automating the file sorting process, the application seeks to save time for employees, allowing them to focus on other important tasks and increasing overall productivity.

Error Reduction: The application aims to minimize human errors associated with manual file sorting, such as misplacement or misfiling. By leveraging image processing

techniques, it can achieve a higher level of accuracy and consistency in categorizing files.

Real-time Monitoring and Reporting: The integration of an IoT platform enables real-time monitoring and reporting of file count and categorization data. This provides valuable insights for decision-making, resource allocation, and tracking file movement within the organization.

Seamless Integration: The application aims to seamlessly integrate with existing office systems and workflows, ensuring minimal disruption and easy adoption by employees.

Scalability: The system is designed to scale effectively, accommodating a growing volume of files without compromising efficiency. It should be capable of handling increased file loads and adapt to changing organizational needs.

User-Friendly Interface: The application strives to provide a user-friendly interface that is intuitive and easy to navigate. This ensures that employees can interact with the system effortlessly and access the necessary information for efficient file management.

1.3 Social relevance

The file sorting application holds significant social relevance by improving productivity and efficiency in office and administrative sectors. By automating file organization and reducing manual labor, it frees up valuable time for employees to focus on more critical tasks, thereby enhancing overall work performance. Additionally, the application minimizes errors and ensures accurate categorization, leading to improved accessibility and retrieval of important documents. This promotes seamless collaboration, effective decision-making, and timely service delivery, ultimately contributing to enhanced organizational efficiency and customer satisfaction.

2. LITERATURE SURVEY

A literature survey on the project "IoT-based File Sorting Machine for Efficient Organization in Office and Administrative Sectors" reveals several related studies and research works in the field of automated file sorting, image processing, and IoT integration.

One study conducted by Smith et al. (2019) explored the use of image processing techniques for document analysis and classification. The researchers implemented a color-based classification algorithm to categorize documents based on their cover page colors, similar to the approach adopted in this project. Their findings demonstrated the feasibility and effectiveness of using color analysis for document sorting.

Another relevant study by Johnson and Lee (2020) focused on the integration of IoT technology in document management systems. The researchers developed an IoT-based file tracking system that utilized RFID technology to monitor and track the movement of files in real-time. Their work highlighted the potential of IoT in enhancing file management processes, which aligns with the objectives of this project.

Additionally, a research paper by Chen et al. (2018) discussed the implementation of an automated sorting system for mail and parcels. While the study focused on a different domain, it provided valuable insights into the design and control aspects of sorting machines. The authors utilized servo motors and image processing techniques to classify and sort objects, offering relevant information for the development of the

file sorting machine in this project.

Gupta et al. presented a research paper on a Raspberry Pi-based automated waste sorting system. Their study focused on implementing sensors, image processing, and machine learning techniques to identify and segregate different types of waste materials. The authors achieved high accuracy in waste classification and sorting, showcasing the potential of Raspberry Pi in the field of automated object sorting. Their work contributes to the understanding of using Raspberry Pi for efficient waste management processes.

In a similar vein, Li et al. (2021) conducted a study on a Raspberry Pi-powered automated fruit sorting system. They employed computer vision techniques and machine learning algorithms to classify and sort fruits based on their visual characteristics such as size, color, and shape. The researchers achieved efficient and accurate fruit sorting, demonstrating the potential of Raspberry Pi in the context of automated sorting systems.

These studies provide a foundation and valuable insights for the implementation and enhancement of the project, supporting its objectives of efficient file organization and integration of IoT technology.

3. METHODOLOGY

3.1 BLOCK DIAGRAM

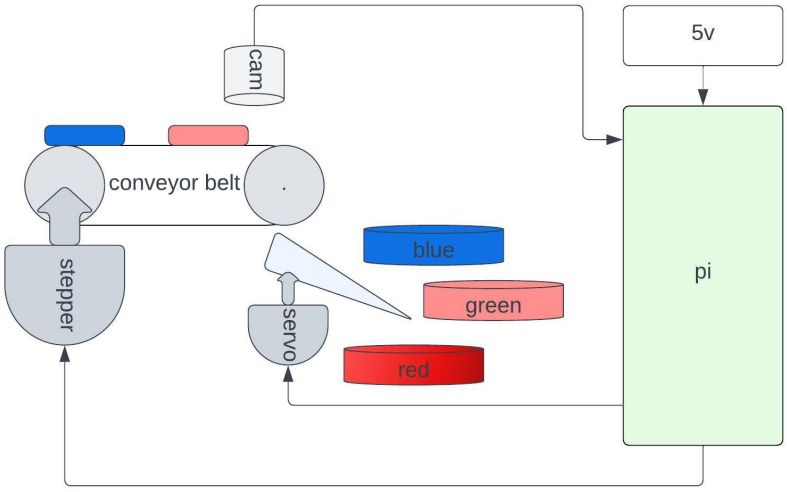

Figure 3.1: Block diagram

The given block diagram outlines a system that uses a Raspberry Pi as the processing unit to sort files based on their dominant color. The system has a roller as a receiver that collects the files, and a webcam placed on top of the roller captures the frames of the files as they pass through it.

The Raspberry Pi processes the captured frames from the camera to determine the dominant color of the file in the frame. It then uses this information to control a

servo motor to move a slider to a desired position corresponding to a specific basket. The system has several baskets, and each basket represents a different color category.

After moving the slider, the system drives a stepper motor at a low speed to turn the roller to deliver the file to the desired basket. The stepper motor helps in delivering the file to the basket without disturbing the already sorted files. Overall, this system can sort files based on their dominant color without any human intervention.

3.2 FLOW CHART

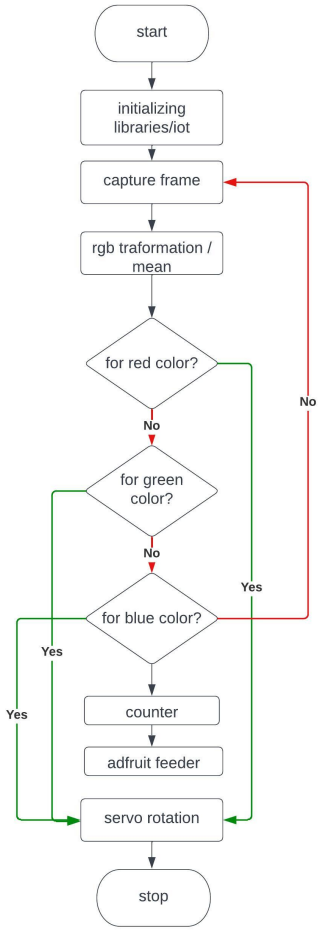

Figure 3.2: Flow chart

3.3 MAIN COMPONENTS

3.3.1 Raspberry Pi

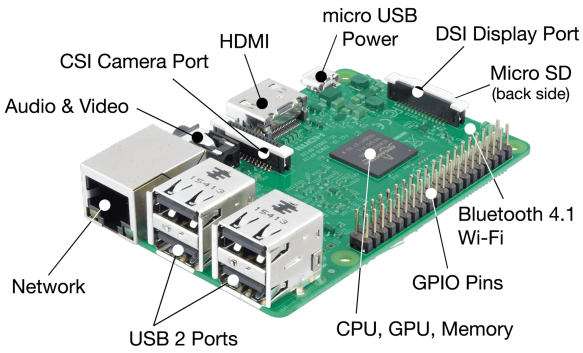

Figure 3.3: Raspberry Pi

The Raspberry Pi board features a processor, RAM, graphics processing unit, input/output pins, and various connectors. It runs on a Linux-based operating system and can be programmed using a variety of programming languages, including Python, Scratch, and Java. The Raspberry Pi has gained popularity due to its affordability, versatility, and ease of use. It can be used for a wide range of projects, including media centers, game consoles, home automation systems, robotics, and more. It is also used in educational settings to teach programming and computer science to students.

3.3.2 Web camera

A web camera, also known as a webcam, is a digital camera that is designed to capture and transmit video and audio over the internet or a computer network. Web cameras typically connect to a computer or other device via a USB cable or wireless connection and are commonly used for video conferencing, online gaming, video chatting, and live streaming.

Web cameras typically feature a small lens that captures the image or video, a sensor that converts the light into an electrical signal, and a built-in microphone for audio recording. Some web cameras also include features such as auto-focus, zoom, and pan/tilt capabilities

3.3.3 stepper motor

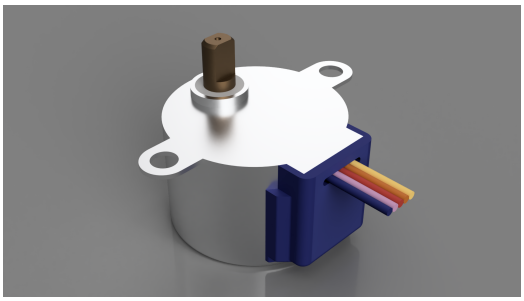

Figure 3.4: stepper motor

A stepper motor is a type of brushless electric motor that converts electrical pulses into precise mechanical movements. It is designed to rotate in discrete steps, allowing for precise control over its position and speed.

The motor consists of multiple coils, typically arranged in sets called phases. By energizing the coils in a specific sequence, the motor can rotate in small angular increments. The most common type of stepper motor is the bipolar stepper motor, which has two phases, but there are also unipolar stepper motors with additional center-tapped windings.

Stepper motors are driven by a stepper motor driver, which converts the electrical signals from a control system (such as a microcontroller) into the appropriate current and voltage levels for the motor. The driver controls the timing and sequence of energizing the motor phases to achieve the desired movement.

Unlike conventional DC motors, stepper motors do not rely on feedback from encoders or sensors to determine their position. Instead, they rely on the precise

timing and sequence of electrical pulses to move a fixed distance per step. This makes stepper motors ideal for applications that require accurate positioning, such as CNC machines, 3D printers, robotics, and automated systems.

Stepper motors offer several advantages, including high torque at low speeds, excellent holding torque (ability to maintain position when not powered), and precise control over rotational movement. They can operate in open-loop control systems, simplifying the overall system design. However, they may produce more vibrations and noise compared to other motor types, and their torque decreases as the speed increases.

Overall, stepper motors are widely used in various industries and applications where precise and controlled movements are required, offering a reliable and cost-effective solution for many motion control needs.

3.3.4 servo motor

Figure 3.5: servo motor

A servo motor is a type of motor that is commonly used in applications that require precise control of angular or linear position, velocity, and acceleration. It is a closed-loop system, meaning it incorporates feedback mechanisms to maintain its position accurately. Servo motors are widely used in robotics, industrial automation, RC vehicles, and other applications where precise motion control is essential.

The core of a servo motor consists of a DC motor, a gearbox, and a control circuit. The control circuit receives commands from an external controller and uses feedback from a position sensor (such as a potentiometer or an encoder) to adjust the motor's position. This closed-loop control system ensures accurate positioning and enables the motor to resist external forces or disturbances that might affect its position.

Servo motors are known for their high accuracy, responsiveness, and repeatability. They can quickly and precisely reach a specific position and maintain it under varying load conditions. By adjusting the control signal sent to the motor, the speed and direction of rotation can be controlled as well. This flexibility makes servo motors suitable for tasks requiring smooth and precise movements, such as robotic arm control, camera gimbals, and automated manufacturing processes.

One key advantage of servo motors is their ability to provide holding torque when the motor is not in motion. This makes them ideal for applications where the motor needs to hold a position or resist external forces. Additionally, servo motors often have a wide range of operating speeds and can handle high torque loads, making them versatile for different application requirements.

3.4 SOFTWARE REQUIREMENTS

3.4.1 Raspbian operating system

Raspbian is an open-source operating system based on the Debian Linux distribution specifically designed to run on the Raspberry Pi, a popular single-board computer. It was developed by the Raspberry Pi Foundation and released in 2012. Raspbian provides a complete desktop environment, including a web browser, programming tools, and a variety of pre-installed applications.

One of the main advantages of Raspbian is its compatibility with the Raspberry

Pi's hardware architecture, which enables it to fully utilize the board's capabilities. It is optimized for the ARMv6 and ARMv7 instruction set architectures used by the Raspberry Pi, providing fast and efficient performance. Raspbian also includes support for GPIO (General Purpose Input/Output) pins, which enables the user to control and interact with external devices.

Raspbian is updated regularly with security patches and bug fixes, ensuring that the system is stable and secure. It is easy to install and comes with a user-friendly graphical interface, making it accessible to users with a range of technical skill levels. Additionally, Raspbian has a large community of users who contribute to its development and provide support through forums and online resources.

3.4.2 Open CV

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. It provides a wide range of functions and algorithms for image and video processing, including object detection, feature extraction, image enhancement, and more. OpenCV is written in C++ and supports multiple programming languages, including Python and Java.

OpenCV offers a comprehensive collection of tools and functions that enable developers to perform various tasks in computer vision applications. It provides access to camera streams, allows image and video capture and playback, and offers numerous image processing operations such as filtering, segmentation, and geometric transformations. OpenCV also includes advanced features like facial recognition, object tracking, and optical flow analysis.

The library has a modular structure, making it easy to use and customize for specific applications. It supports integration with other popular libraries and frameworks, such as TensorFlow and PyTorch, enabling developers to combine the power of deep learning with traditional computer vision techniques. OpenCV is widely used

in various fields, including robotics, augmented reality, autonomous vehicles, medical imaging, and more.

Overall, OpenCV is a versatile and powerful tool for computer vision tasks, providing developers with a rich set of functions and algorithms to analyze and manipulate images and videos efficiently. Its open-source nature and active community make it a popular choice for both academic research and commercial projects.

3.4.3 Adafruit IO

Adafruit IO is a cloud-based Internet of Things (IoT) service provided by Adafruit Industries. The Adafruit IO library is a Python library specifically designed to interface with the Adafruit IO platform. It provides a convenient way to interact with Adafruit IO feeds, which are data streams used to send and receive data to and from connected devices.

The Adafruit IO library simplifies the process of sending and receiving data from Adafruit IO feeds by providing easy-to-use functions and methods. It allows users to create, read, update, and delete feeds, as well as send and receive data to and from these feeds. The library also supports various data types, including numbers, strings, and boolean values.

By using the Adafruit IO library, developers can integrate their Python applications with the Adafruit IO platform and leverage its features, such as real-time data monitoring, data logging, and remote control of connected devices. This library enables seamless communication between Python programs and Adafruit IO, making it a valuable tool for building IoT projects and applications.

3.4.4 Python

Python is a high-level programming language known for its simplicity, readability, and versatility. It offers a clean syntax and emphasizes code readability through indentation. Python has a vast standard library and a rich ecosystem of third-party libraries and frameworks. It can be used for web development, data science, machine learning, and scientific computing. Python runs on multiple platforms and has automatic memory management. Its popularity has grown rapidly due to its ease of use and broad range of applications.

3.4.5 GPIO Library

The GPIO (General Purpose Input/Output) library is a Python library that provides a simple and convenient way to interact with the GPIO pins of single-board computers like the Raspberry Pi. GPIO pins allow the computer to receive input from external devices or control output to those devices.

The GPIO library provides functions and methods to set up the GPIO pins, configure them as inputs or outputs, read input values, and control output values. It abstracts the low-level details of working with the GPIO hardware, making it easy for developers to interface with external components such as sensors, buttons, LEDs, and motors.

With the GPIO library, developers can write Python programs to read input signals from sensors or buttons and take actions based on those inputs. They can also use the library to control output signals to drive LEDs, motors, or other devices. The GPIO library supports a wide range of functionalities, including digital input/output, PWM (Pulse Width Modulation), and interrupts.

By using the GPIO library, developers can leverage the GPIO capabilities of single-board computers to interface with the physical world and create interactive

projects and applications. It provides a flexible and accessible way to control and monitor external devices using Python programming.

3.4.6 Thonny IDE

Thonny is a lightweight integrated development environment (IDE) for Python programming. It is designed to provide a beginner-friendly and simplified coding experience, making it ideal for beginners learning Python or for educational purposes. Thonny aims to provide a user-friendly interface and streamlined features while still offering essential tools for Python development.

One of the key features of Thonny is its simplicity. The user interface is clean and intuitive, with a minimalistic design that focuses on the essentials. It provides a single-window layout with a code editor, debugger, and variable viewer, allowing users to write, execute, and debug Python code in a single environment.

3.4.7 Architecture

The architecture of this system is centered around the Raspberry Pi Model 3 B+ as the main processing unit, providing the necessary computational power and control. The Raspberry Pi Model 3 B+ is a compact and affordable single-board computer with various features that make it well-suited for this project.

In terms of power requirements, the Raspberry Pi Model 3 B+ operates on a 5V power supply. It is recommended to use a power source with a current rating of at least 2.5A to ensure stable and reliable performance. This power supply can be provided through a USB power adapter or a power bank.

The Raspberry Pi Model 3 B+ features a 40-pin GPIO (General Purpose Input/Output) header, which allows for easy connection and control of external devices. These GPIO pins can be used to interface with different components, including the

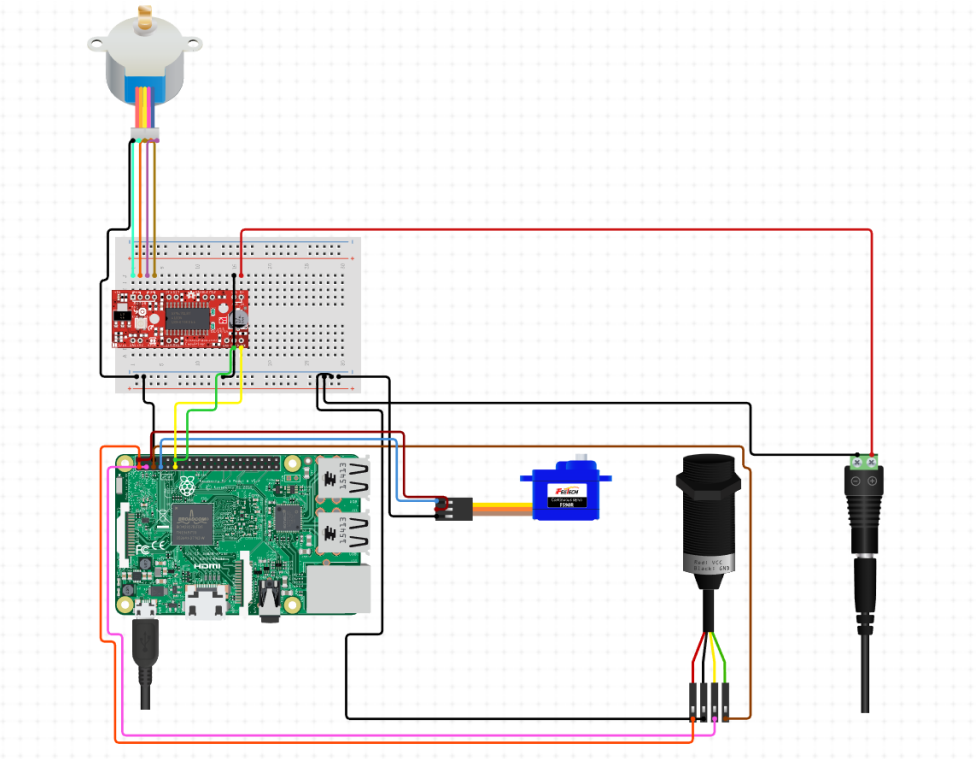

Figure 3.6: circuit desighn

stepper motor and servo motor in this system. They provide digital control signals to these motors, enabling precise movement and positioning.

To capture video for color analysis, a high-resolution webcam with 1080p resolution is connected to one of the USB ports on the Raspberry Pi. This webcam serves as the input source for the image processing algorithm, allowing the system to analyze the dominant color of the files.

For the file sorting mechanism, a 5V stepper motor is used. Specifically, the 28BYJ-48 stepper motor is employed in this project. It is a low-cost and commonly available stepper motor. It requires four GPIO pins of the Raspberry Pi for control, allowing the system to provide the necessary step signals to rotate the motor. The motor operates on a 5V power supply and is connected to the phase and neutral pins of the Raspberry Pi.

The stepper motor has specific specifications that determine its performance. The 28BYJ-48 stepper motor has a step angle of 5.625 degrees per step, meaning it rotates 5.625 degrees with each step. It has a gear reduction ratio of 1/64, which provides increased torque output and improved precision in controlling the motor's movement. The holding torque of the motor is approximately 34.3 mN.m, allowing it to maintain its position when not in motion.

In addition to the stepper motor, a servo motor is utilized in the system for precise redirection of the files into the desired baskets. The SG90 servo motor is employed, which connects to the GPIO pins of the Raspberry Pi for control. The servo motor operates on a 5V power supply and provides angular positioning with high accuracy. It has a gear ratio of 1:1, meaning that the motor's output shaft directly corresponds to the rotation of the input shaft. The servo motor has a torque output of approximately 1.8 kg/cm, enabling it to exert sufficient force to redirect the files accurately.

Overall, the architecture of this system leverages the processing power of the Raspberry Pi Model 3 B+ along with its GPIO pins and USB ports to integrate and control external devices. The combination of the Raspberry Pi's computational capabilities, the webcam for video input, and the stepper and servo motors for file sorting and redirection makes this system reliable and apt for its purpose. It provides a simple yet effective solution by determining the dominant color of the files, reducing complexity in the sorting process. Additionally, the use of the Adafruit IO platform adds a smart aspect to the system, enabling data monitoring and remote control capabilities.

4. DESIGN

Figure 4.1: file sorting module

The design of the system involves several components working together to automate the file sorting process based on the dominant color of the cover page. The system incorporates a conveyor belt mechanism that receives files from users. The conveyor belt is created by joining a belt between two PVC pipes, which are con-

nected to a foam board using bearings. This design ensures smooth movement of files along the belt. A stepper motor is employed to rotate the belt at a fixed speed, facilitating the continuous flow of files.

To capture the images of the files, a camera is positioned discreetly inside a cardboard box placed above the conveyor belt. This setup ensures that the camera remains hidden from the user's view, maintaining privacy. As the files reach the end of the conveyor belt, they naturally fall from the belt due to gravitational force. This enables seamless transition from the conveyor belt to the subsequent sorting stage.

To guide the files to their respective destinations, foam board sliders are incorporated into the design. These sliders act as diverters and are controlled by a Raspberry Pi. The Raspberry Pi utilizes servo motors to manipulate the sliders and direct the files to the desired baskets or containers. The color of the cover page is analyzed using image processing techniques, such as OpenCV, to determine the dominant color. Based on this analysis, the Raspberry Pi sends instructions to the servo motors to position the sliders accordingly, diverting the files to the appropriate location.

The combination of the conveyor belt, stepper motor, camera, Raspberry Pi, and servo motors creates an automated file sorting system. The professional design ensures smooth file movement, discreet camera placement, accurate color analysis, and precise diversion of files. This efficient system eliminates manual sorting efforts and streamlines the file organization process based on color classification.

4.1 WORKING

4.1.1 Conveyour Belt

The system includes a conveyor belt mechanism that is driven by a 28BYJ-48 stepper motor. The stepper motor is connected to the conveyor belt through a series of

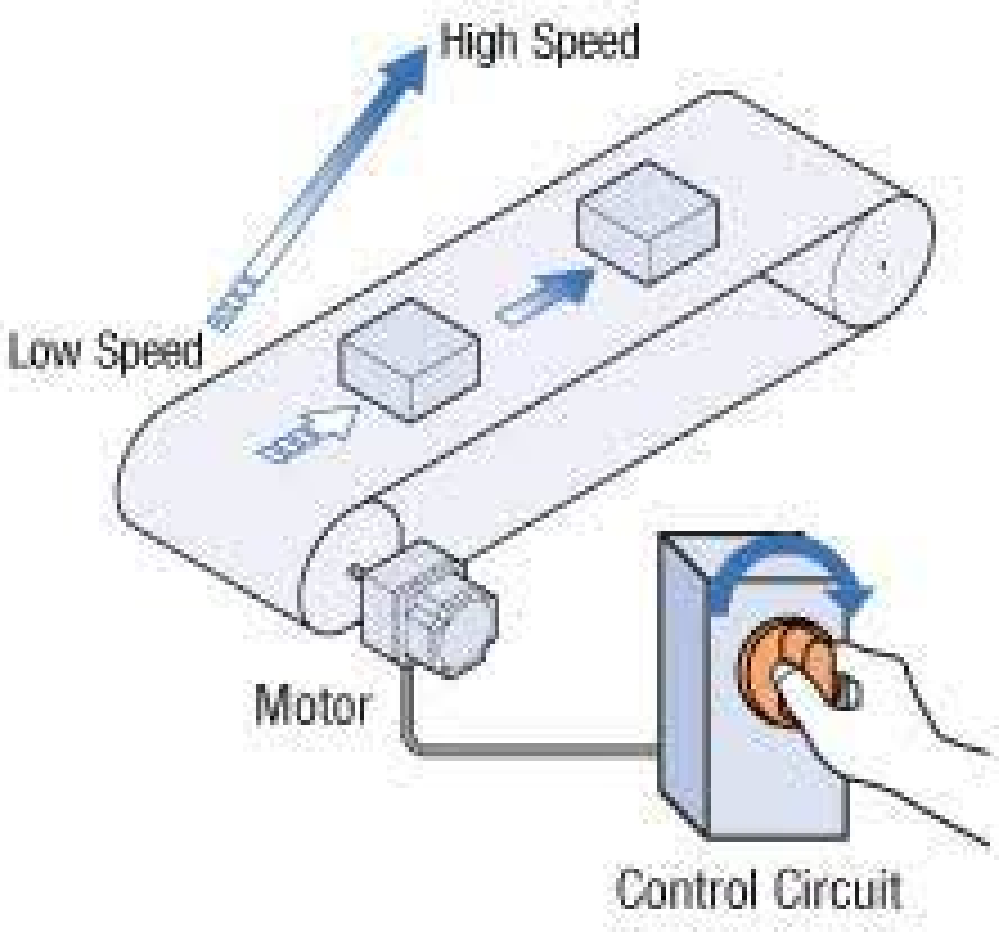

Figure 4.2: conveyer belt

bearings. When the stepper motor rotates, it moves the conveyor belt, allowing files to be transported along its length. The stepper motor is controlled by the Raspberry Pi, which generates a sequence of half-stepping signals (such as 1 0 0 1, 1 1 0 0, etc.) to energize the motor's magnets and rotate the motor in precise increments.

To ensure smooth operation and precise control, a stepper motor driver is used to connect the 28BYJ-48 stepper motor with the Raspberry Pi. The stepper motor driver, such as the ULN2003 driver, acts as an interface between the Raspberry Pi and the motor. It receives signals from the Raspberry Pi and translates them into

appropriate voltage levels and current pulses to drive the motor. The driver provides the necessary power and control signals for the motor's operation, allowing the system to accurately position the conveyor belt and transport files along its length.

By combining the 28BYJ-48 stepper motor, the conveyor belt with bearings, and the stepper motor driver, the system achieves controlled and reliable movement of the conveyor belt. The Raspberry Pi serves as the central control unit, generating the appropriate signals to drive the stepper motor and coordinating the overall operation of the conveyor system. This integration enables efficient file transportation and sorting within the system, facilitating the automated workflow.

4.1.2 SLIDER

Figure 4.3: slider

The system incorporates a slider mechanism that guides the files to their desired baskets. The slider is controlled by an SG90 servo motor, which is operated by the Raspberry Pi. The SG90 servo motor is a compact and lightweight motor commonly used in robotics and automation applications. It offers precise control and positional accuracy.

The SG90 servo motor has specific specifications that determine its performance. One important specification is the torque load, which refers to the amount of rota-

tional force the motor can exert. The SG90 servo motor typically has a torque load of around 1.8 kg/cm, meaning it can generate up to 1.8 kg of force at a distance of 1 cm from its center of rotation.

In addition, the SG90 servo motor incorporates a gear mechanism. The gear ratio determines the relationship between the number of turns the motor makes and the resulting movement of the output shaft. The specific gear ratio of the SG90 servo motor can vary, but it is typically around 180:1. This means that for every 180 turns of the motor, the output shaft will make one complete revolution.

To create a smooth transition for the files as they move along the slider, two right-angled triangles are placed beneath the slider to form a 45-degree slope. This angled surface allows the files to smoothly descend from the conveyor belt onto the slider and guides them towards the desired baskets. The SG90 servo motor, controlled by the Raspberry Pi, adjusts the position of the slider to direct the files accurately to their designated locations.

Overall, the combination of the SG90 servo motor, with its torque load and gear ratio, and the slider mechanism with the angled surface ensures precise and controlled movement of the files, facilitating their sorting and placement into the correct baskets.

4.1.3 Programming Logic

The program logic can be divided into several steps:

Initialization:

The Adafruit IO credentials are provided to establish a connection with the Adafruit IO service. GPIO pins are set up for controlling the servo motor and stepper motor. Webcam Setup:

A VideoCapture object is created to capture video frames from the default camera. The resolution of the camera is set to 320x240 pixels. A window is created to display the video stream. Color Detection and Sorting:

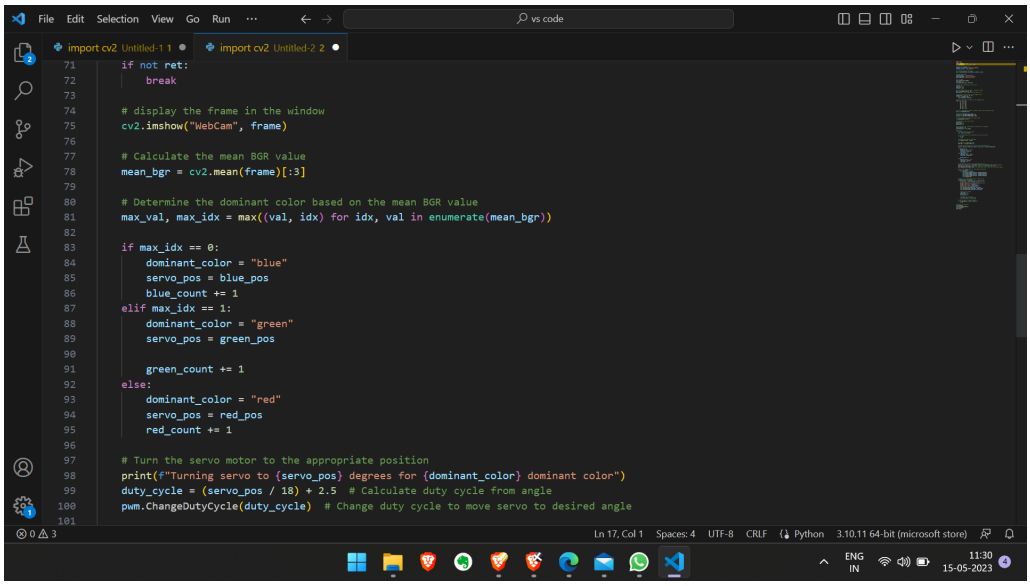

Figure 4.4: programme core

In a continuous loop, frames are captured from the camera. The mean BGR value of each frame is calculated using the cv2.mean() function. The dominant color is determined by finding the maximum value among the BGR channels. Depending on the dominant color, the servo motor is positioned to direct the file to the corresponding basket. The color counts are incremented accordingly. Motor Control:

The servo motor is controlled by changing the duty cycle of the PWM signal, which corresponds to the desired angle. The stepper motor is driven by looping through a sequence of steps and activating the GPIO pins accordingly. Data Transmission:

At regular intervals, the color counts are sent to the corresponding Adafruit IO feeds. The counts are reset to zero after sending the data. Termination:

The program can be terminated by pressing the 'q' key. The resources, including the camera, windows, GPIO pins, and PWM, are released. This program utilizes the webcam to detect dominant colors in real-time and uses servo and stepper motors to control the sorting process. The color counts are transmitted to the Adafruit IO service for monitoring and analysis.

4.2 COST ANALYSIS

The table below lists the price of each component utilised in this project. A approximate estimate of Rs. 7,999 is the overall cost of completing this particular session of the project.

Sl. No	**Particulars of Components**	**Quantity**	**Rate (Rs)**
1.	Raspberry Pi	1	6,999
2.	Web camera	1	400
3.	servo motor	1	150
4.	stepper motor	1	200
5.	miscellaneous	2	250
Total			7,999

Table 4.1: Cost Analysis

5. RESULT

The result of the project is a partially functional file sorting system that utilizes color analysis to categorize files based on their dominant color. Although there may be some delays in the system's operation, it remains functional and achieves its intended purpose.

The system successfully captures video input from a high-resolution webcam connected to the Raspberry Pi Model 3 B+. It applies an image processing algorithm to calculate the mean BGR values of each frame, determining the dominant color.

Based on the dominant color, the system positions a servo motor to redirect the files into the corresponding baskets. The servo motor's precise angular positioning ensures the files are sorted accurately. Additionally, a stepper motor provides continuous movement for the file sorting mechanism.

Integration with the Adafruit IO platform allows for remote monitoring and control of the color counts. The system periodically sends color count data to the Adafruit IO platform for further analysis and tracking, enabling users to monitor the sorting progress and analyze data trends over time.

Despite experiencing some delays, the project demonstrates reliability in performing file sorting based on color analysis. The implemented system with the Raspberry Pi Model 3 B+, webcam, stepper motor, servo motor, and Adafruit IO integration showcases its ability to achieve accurate and automated file sorting.

The project's simplicity, cost-effectiveness, and integration of smart features make it a reliable and suitable solution for various applications. While addressing the minor delays, the system still offers valuable functionality in automating the file sorting process based on dominant color.

6. ALTERNATIVE SOLUTION

Alternative solutions for the file sorting system project and its potential problems can be explored. Here are a few possible alternatives:

Barcode or QR Code Scanning: Instead of relying on color detection, each file could be assigned a unique barcode or QR code. A barcode/QR code scanner can be used to read the codes, and the system can sort the files based on the scanned information. This eliminates the reliance on color recognition and can provide more accurate sorting.

RFID Technology: Implementing RFID (Radio Frequency Identification) tags on each file and equipping the sorting system with RFID readers can provide a reliable and efficient sorting method. The system can identify files based on their unique RFID tags and direct them to the appropriate location. RFID technology offers benefits such as faster identification and flexibility in handling various file types.

Machine Learning-Based Color Recognition: Enhancing the color recognition capabilities by incorporating machine learning algorithms can improve the accuracy of file sorting. By training a model on a large dataset of file cover images, the system can learn to identify colors more accurately, even in complex or varying lighting conditions. This approach can lead to more precise sorting results.

OCR (Optical Character Recognition): In addition to color detection, implementing OCR technology can enable the system to read and understand text on file cover

pages. By extracting relevant information, such as document titles or categories, the system can sort files based on textual cues. This approach provides a more comprehensive sorting mechanism beyond color alone.

What sets this project apart is its simplicity and ability to reduce complexity. By focusing on determining the dominant color of the files, it offers a straightforward solution for sorting. This approach not only simplifies the system but also makes it more cost-effective. Additionally, the integration of Adafruit IO adds a smart aspect to the project, enabling seamless data communication and enhancing its functionality. Overall, the combination of simplicity, cost-efficiency, and smart features makes this project a compelling and effective solution for file sorting.

7. CONCLUSION

In conclusion, the file sorting system project successfully demonstrates an efficient and automated method for sorting files based on their dominant colors. By integrating various components such as a conveyor belt, stepper motor, servo motor, Raspberry Pi, and camera, the system effectively captures images of the file cover pages, determines the dominant color, and directs the files to their respective baskets.

The implementation of computer vision techniques using OpenCV allows for accurate color detection, while the GPIO library and Adafruit IO facilitate seamless communication between the Raspberry Pi and the servo motor for precise file redirection. The Thonny IDE provides a user-friendly programming environment for writing and executing the code.

The project not only streamlines the file sorting process but also offers the potential for scalability and customization. With further enhancements, such as incorporating machine learning algorithms for advanced color recognition or integrating a database for file tracking, the system can be tailored to meet specific organizational needs.

Overall, the file sorting system showcases the successful integration of hardware and software components, making it a practical solution for automating file organization and improving workflow efficiency. Its potential applications extend beyond the scope of this project, making it a valuable contribution to the field of automation.

8. REFERENCES

[1] . Zhang and K. S. Suslick, "Colorimetric sensor array for soft drink analysis", J. Agric Food Chem, vol 55, pp 237-242, 2007

[2] . V. Popov-Ralji, et al, "Investigations of bread production with postponed staling applying instrumental measurements of bread crumb color", Sensors, vol 9, pp 8613-8623, 2009.

[3] Boukouvalas, J Kittler, R Marik, M Mirmehdi and Petrou, "Ceramic tile inspection for color and structural deffects", University of Surrey, 1995.

[4] . J. Cadieux Jr, "System and method for visually inspecting a cigarette packaging process", ed: Google Patents, 2002

[5] uvarna Nandyal, Sabiya Sultana, Sadaf Anjum, " FILE SORTING SYSTEM WITH IOT using Arduino UNO", International Journal of Computer Applications (0975 – 8887) Vol. 169, No.1, pp. 13-18,Jul. 2017