Interim Report

Shaheen Amin, 01866464 Dr. Edward Stott, Supervisor

December 2023

Abstract & Plagirism Declaration

To address the issue of electronic waste and cluttered workspaces in the Level 1 Labs, this project aims to develop an automated system for identifying and sorting electronic components. Utilizing Computer Vision, the system will classify components like resistors, capacitors, ICs, and MOSFETs, and sort them into designated bins. The project will be executed in three stages: initial development of a Computer Vision system for component identification, integration of a semi-automated sorting mechanism, and potentially, full automation of the sorting process or additional features like IC testing. Supervised by Dr. Edward Stott, this project aims to practically solve the problems faced by the Level 1 Labs.

I affirm that I have provided explicit references for all the material in my Interim Report that is not authored by me, but is represented as my own work. I have used GPTv4, as an aid in the preparation of my report. It was used for LaTeX formatting, however all technical content is original.

Contents

1 Project Specification

This project aims to solve the problem of electronic waste and cluttered workspaces in the Level 1 Labs by developing an automated system for identifying and sorting electronic components.

It aims to do so by employing state of the art Computer Vision techniques to classify various electronic components, and then sort them into designated bins. The project's deliverables are as follows:

- A Computer Vision system for component identification
- An interface from which to observe and control the system's state
- A semi-automated sorting mechanism
- Potentially, full automation of the sorting process or additional features like IC testing

2 Background

For the background of this project, it is necessary to consider literature related to the following:

- Mechanical design of existing sorting machines
- Existing Computer Vision systems for component identification
- Real-time Computer Vision architectures

2.1 Mechanical Design

Bowl Feeders

In my research, I have discovered that most

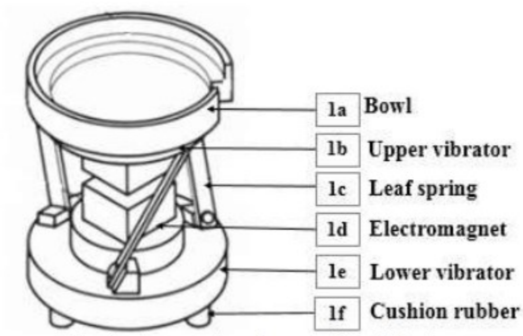

Figure 1: VBF Nam et al.[10]

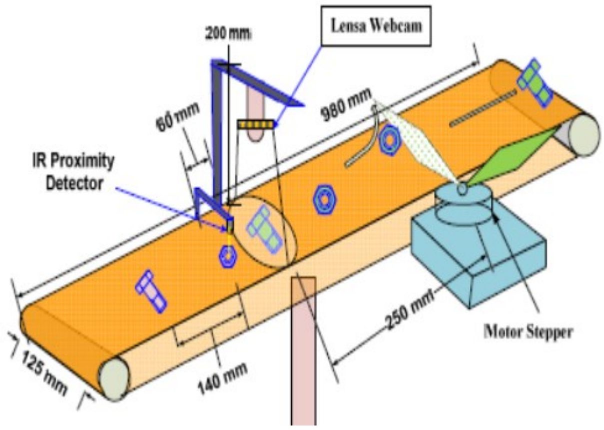

Figure 2: Conveyer Belt Dhenge et al.[4]

industrial sorting machines use a Vibratory Bowl Feeder (VBF) to feed components into the system; as shown in Figure 1, the VBF consists of a bowl that vibrates coupled with a spring and electromagnet to feed components into the system. The paper by Nam et al.[10] explores the optimal design of a VBF for USB keycaps, by attempting to identify the ideal parameters for the structure of the bowl, sorting track, mounting adapter, and suspension system. The paper also uses modal analysis to determine the natural frequencies of the system, and uses this to avoid resonant conditions that might cause inefficient or erratic operation.

This paper is useful as it provides a comprehsive overview of the design of a VBF, and provides a good starting point for the design of the VBF for in the future when the project is extended to include a VBF for fully autonomous sorting. The paper also provides a good overview of the design considerations for the VBF, and so can be used as a reference during the design process.

Additionally, Reinhart and Loy[16] delves into the mathematical modelling of a VBF, focusing more on the overall performance of the VBF rather than efficiency, and Silversides et al.[17] provides a good overview of the forces involved in the operation of a VBF, stengthening the basis for its design and viability.

The paper by Zhang and of Technology. Department of Mechanical Engineering[19] outlines a sorting system for vials, and does not make use of a VBF, instead opting for a turntable design that mechanically orientates the vials. It primarily operates by using a design that is specific to the geometry of the vials, and so is not applicable to this project, however it does provide a good insight into the design of a sorting system.

It seems conclusive that a VBF is the most viable option for the feeding mechanism of the sorting system, and so the design of the VBF will be based on the research outlined above.

Transport Mechanisms
The paper by Dhenge et al.[4]

3 Implementation

The project is to be executed in three stages:

- Initial development of a Computer Vision system for component identification
- Integration of a semi-automated sorting mechanism
- Potentially, full automation of the sorting process or additional features like IC testing

This approach was taken to allow me to set defined goals to ensure that the project is always working towards a deliverable product. Each stage extends the previous stage, and so the project can be considered as a series of smaller projects.

For this initial stage in the project, the focus is on developing the foundation for the system to be built upon so that the development of the various systems can be done independently, in parallel, and in such a way that each system is modular and easily extensible. I have identified four systems that compromise the project:

- The Hardware, Mechanics and Electronics
- The software that coordinates the various components
- The Computer Vision system
- The tools used to develop the system

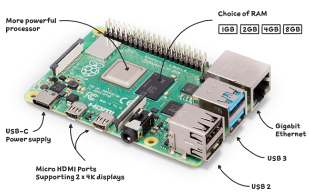

Figure 3: Raspberry Pi 4 Model B 2GB[5]

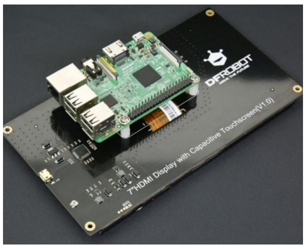

Figure 4: DFRobot 7″ Touchscreen Display[3]

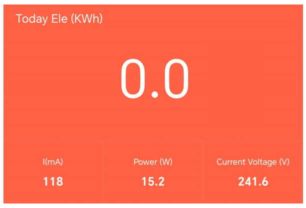

Figure 5: Power consumption of the system

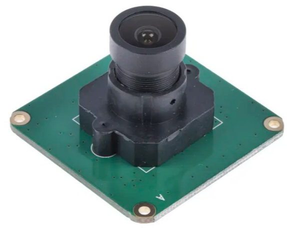

Figure 6: Okdo Adjustable Focus OV5647 Camera[11]

3.1 Hardware, Mechanics and Electronics

The hardware, mechanics and electronics (HME) of the system are the physical components that make up the system. It is necassary to first consider the HME of the system as they are the foundation upon which the rest of the system is built; for example, when designing the Computer Vision system, it should utilise training data that was collected using the HME that the system will be deployed on, ensuring that the system is robust to the conditions it will be used in. As such, the main focus of this stage has been the HME.

3.1.1 Hardware

For the initial stage of the project, the main hardware components of the HME are:

- Raspberry Pi 4 Model B 2GB
- 7″ Touchscreen Display
- 24V dc, 6.25A, 150W Power Supply
- Okdo Adjustable Focus OV5647 Camera
- LED Light Strip

Raspberry Pi 4 Model B 2GB

The Raspberry Pi 4 Model B is the main component of the system, and is the central hub that all other components are connected to. This model was chosen as it is regarded as a reliable SoC and is widely used in the industry. It has a large amount of software and driver support, so I can be confident that I will be able to find a solution to any issues that may arise. Additionally, it has GPIO pins that can be used to control other components which will be necassary in future stages. It also has a dedicated camera port which allows for a camera to be connected directly to the SoC, which is necassary for the Computer Vision system.

It also has WiFi support for SSH and remote development, as well as HDMI port for the display. Originally, the 4GB model was ordered however an issue with the EE Stores resulted in receiving the 2GB model instead. This was not an issue as the 2GB model is sufficient for the initial stage of the project, however if it is found to be a bottleneck in future stages, a more powerful model may be used. The decision to not return the 2GB model is because the EEStores are not available over the Christmas break.

3.1.2 Mechanics

7″ Touchscreen Display

The DFRobot 7″ Touchscreen Display was chosen as it is a relatively cheap display that is compatible with the Raspberry Pi. It has touchscreen support and has a Raspberry Pi 4 mount on its back, as well as HDMI adapter boards to connect to the Pi. This means I do not have to design a physical mount for the Pi, and only need to design a mount for the display.

24V dc, 6.25A, 150W Power Supply

There are two parameters that were taken into consideration when choosing a power supply: the power output and the voltage.

- **Power Rating**
  The power supply must be able to drive all components in the system with overhead in case of spikes in power consumption. In this first stage, the system only consists of the Raspberry Pi, the display, LED lights and camera, making for a power consumption of under 20W, recorded using a smart plug with a power meter. However, in the future, the system will likely need to drive additional motors and sensors, so the total power consumption will be higher. Even with accounting for this, the power consumption will be under 100W, so a 150W power supply will be sufficient, allowing a 50W overhead. In hindsight, this may be

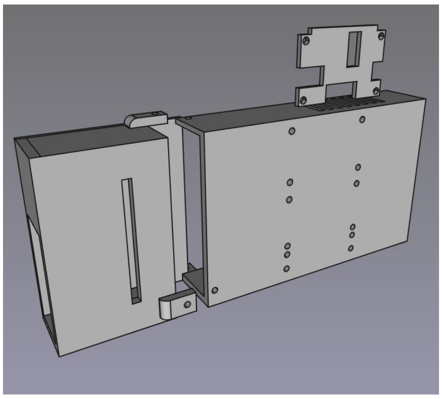

Figure 7: PSU Housing

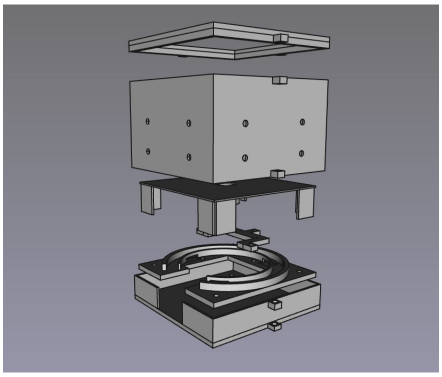

Figure 8: Camera Housing

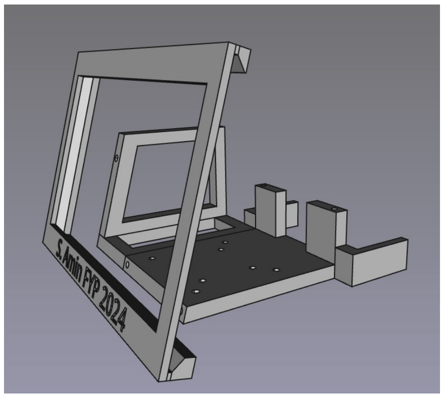

Figure 9: LCD Display Housing

considered excessive.

- **Voltage Rating**
  The voltage of the power supply must be greater or equal than the highest voltage of any component in the system; this is necassary as I have only step-down converters available on hand, so choosing a higher voltage allows me to step-down the voltage to the required voltage for each component. 24V was chosen as it is a common voltage for stepper motors, which may be used in future stages of the project. It is also used for some LED strips, which are used in this stage. The Pi 4 uses 5V and the LED strip uses 12V, which the 24V power supply can be stepped down to.

Okdo Adjustable Focus OV5647 Camera
For the Computer Vision system, a camera is required to capture images of the components. The Okdo Adjustable Focus OV5647 Camera was chosen as it is CSI compatible, meaning it can be connected directly to the Raspberry Pi, and has a manual focus ring, allowing it to be used as a macro camera to capture images of small components placed directly above it. The manual focus ring allows the camera to be specifically tuned to the design of the system, allowing for the best possible image quality. It also has a 5MP sensor[12], which is sufficient for the Computer Vision system, as high resolution images would be preprocessed and reduced.

LED Light Strip
An LED strip is required to illuminate the components so that the camera can capture images of them. Initially, a 12V LED Ring was chosen as it would allow for a uniform light source and can be mounted directly below the camera; however for reasons explained in the next section, it was replaced with a 5V WS2812B LED strip as it solves the issues faced with the LED Ring.

To house the components, a 3D printed housing was designed. It features a modular design, allowing for an iterative design process where I am able to work on different components of the system independently. It also allows for easy replacement of components should my design prove to be inadequate. The housing was designed in FreeCAD[6], a free and open source parametric CAD software that I have personal experience with.
The parametric design allows for easy modification of the design should I need to make changes - with good design practices, I can ensure that changes can be done as I please by simply modifying a few numerical parameters. The design was then printed using my own personal 3D printer, a heavily modified Voxelab Aquila C2. For this stage on the project, I must design 3 major components for the housing:

PSU Housing
The PSU housing contains the power supply and ensures that all high voltage components are safely enclosed. It also houses the power switch, the power socket and the terminals. As I will require step down converters for the Pi and LED strip, the PSU housing also contains mounting points for the step down converters, as well as a mounting plate to ensure that the step down converters are mounted flat to the PSU housing.

Camera Housing
The camera housing contains the camera and the LED strip. It also contains the mounting points for the camera and LED strip. The camera housing will also mount an acrylic plate above the camera, so components can be placed on the plate and be imaged by the camera.
From bottom to top in Figure 8, is the bottom casing, LED Ring mount, camera mount, light

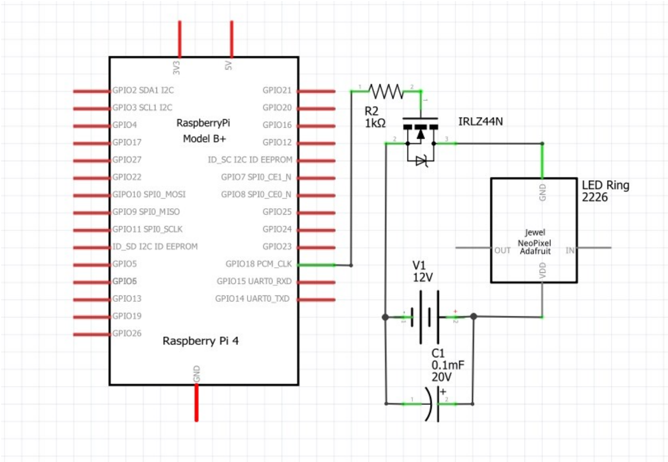

Figure 10: Wiring Schematic

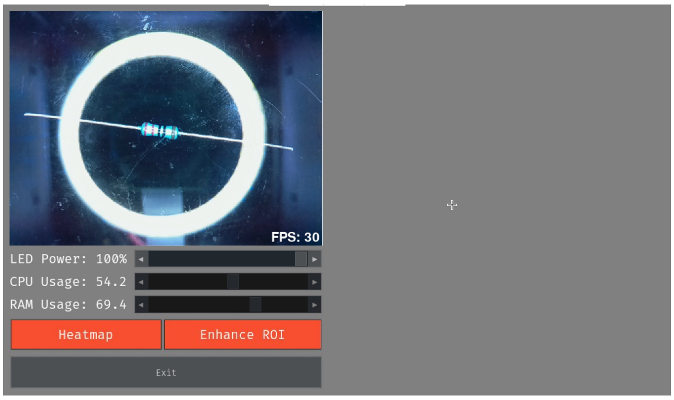

Figure 11: Main UI

diffuser, middle casing and top casing.

LCD Display Housing

The LCD display housing contains the LCD display and the Raspberry Pi. As the 7″ DFRobot LCD display has a Raspberry Pi 4 mount on its back, an explicit mount for the Pi is not required. The mount will position the LCD display at an angle, so it can be viewed from above. The LCD display simply slides into the mount, allowing for easy removal and also contains mounting points to secure the LCD display to the camera housing. The LCD housing is split into two parts; the LCD cover, where the LCD display slides into, and a base plate, which mounts to the camera housing. The LCD cover is then mounted onto the base plate, securing the LCD display in place.

The components are secured using brass M3 heat-set inserts and M3 screws. The heat-set inserts are inserted into the 3D printed parts using a soldering iron, and the components are then screwed into the inserts. This allows for easy removal of components should I need to make changes to the design, following my modular design principle.

3.1.3 Electronics (and Safety)

As my project is a physical system, there are several electronic components that are required to make the system work.

Power Supply

As mentioned in Sections 3.1.1 and 3.1.2, the system is powered by a 24V, 6.25A, 150W power supply, and must be stepped down to 5V for the Pi and 12V for the LED strip. This is achieved using a XL4015 Step Down Converter[8], a variable step down converter that can output up to 75W. The high power rating of the step down converter allows for a large overhead, which is useful for future stages of the project where additional components may be added to the system.

For powering the system, I have chosen a standard IEC C14 power connector and a RS Pro Snap-In Fused Rocker Switch[2]. As the power supply is a 6.25A power supply, I have chosen a 6A fuse for the switch, which is the closest fuse rating to the power supply's current rating. This ensures that the fuse will blow before the power supply is damaged, should there be a short circuit in the system.

For wiring the power supply and connecting the step down converters, I have chosen 18AWG wire, which is rated for 17A[1], more than enough for the system's power consumption. The wire is securely crimped to the power supply using a crimping tool.

Raspberry Pi 4

To power the Raspberry Pi, I have chosen a standard USB-A to USB-C cable and connected it to a female USB-A breakout board connected to the 5V step down converter. This allows for easy removal of the Pi should I need to make changes to the system, and prevents the unnecessary cutting of cables.

As the Pi is mounted to the LCD display, a USB-A to Micro USB cable is used to connect the Pi to the display, which is then connected to the Pi's HDMI port, using a supplied HDMI adapter board.

LED Lighting

As mentioned in Section 3.1.1, initially, a 12V LED Light Ring was used and was changed to be a WS2812B 5V LED strip, for issues described later. As shown in Figure 10, the LED Ring was connected to a 12V supply (achieved using the

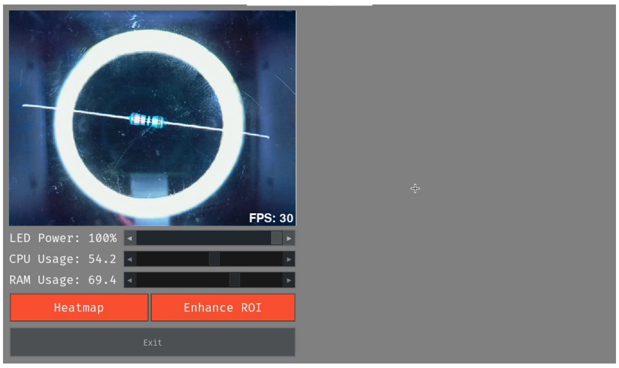

Figure 12: Main UI

12V step down converter) and parallel to a 100μF capacitor, which is used to smooth out the voltage spikes.

The LED Ring was then connected to an N-channel MOSFET IRLZ44N[7], which is controlled by a PWM-enabled GPIO pin on the Pi. This GPIO pin is connected to a 1kΩ resistor, which prevents potential overcurrent from damaging the Pi. The MOSFET is then connected to ground, completing the circuit.

The IRFZ44N is a logic level MOSFET, meaning it can be fully turned on with a 3.3V gate voltage, which is the voltage of the Pi's GPIO pins, and the high switching speed (up to 1mHz) of the MOSFET allows for the LED Ring to be PWM controlled, allowing for the brightness of the LED Ring to be controlled.

3.2 Software

The entire system is controlled by a Raspberry Pi 4B running a very light weight Linux distribution called DietPi. I aim for the entire system to run locally on the Pi, and so the Pi will be responsible for running the Computer Vision system, controlling the hardware, and running the user interface. In order to achieve this, I have identified the following software components:

- The User Interface
- The Computer Vision system
- The Hardware Control system

The system's software is written in Python[15], as it is a language I am very competent with and is well suited for the project, given the availability of libraries for Computer Vision and hardware control. Again, the concept of modularity underpins the design of the software, and so each component is designed to be independent of the others, and can be developed in parallel.

All written Python code adheres to a style defined in a .pylintrc file, which is a configuration file for the pylint[14] linter. This ensures that the code is consistent and readable, and also helps to identify potential bugs and errors. The code is also documented using docstrings for readability and maintainability. Each software component is designed to be self-contained so they can be run independently for testing purposes.

All software constants are defined in a constants.py file, which is imported by all other Python files, allowing for easy modification of constants that modify the behaviour of the system which is useful for testing and debugging.

The User Interface

As the system is intended to be self-contained, it is important to have a user interface that displays all relevant information to the user and allows them to interact with the system. The UI is displayed on the 7″ touchscreen display, and is written using Pygame[13];

This design decision was made after experimentation with other UI libraries like Tkinter[18], however it was found that Pygame was the most viable for several reasons; it contains a module for camera streaming, which is used to display the camera feed on the UI, and can also be used to feed the camera frames to the Computer Vision system; there is a very large amount of control provided by Pygame over the UI, especially in terms of performance. This is crucial as the UI is only there to provide information to the user, and so it is important that it does not interfere with the Computer Vision system or the hardware control system. With an additional library named pygame_gui[9], it is possible to create a UI with GUI elements like buttons, text boxes, and drop down menus, making for a fully comprehensive UI library.

In Figure 11, the current version of the UI is shown. On the left, the camera feed and camera frame rate is displayed, with system CPU and RAM usage displayed, for monitoring purposes. There is a slider to adjust the power of the connected LED Ring light, as well as two buttons related to the Computer Vision system.

The right side is blank for now, but will be used to display information about the Computer Vision system and the hardware control system.

3.3 Computer Vision

3.4 Design Problems and Solutions

3D printing is typically an iterative process, where the design is printed, tested, and then modified based on the results. This further reinforces the need for a modular design, as it allows for easy modification of each component. After a few iterations with errors due to tolerances and slightly incorrect measurements, (not relevant to the technical content of this report) I was able to produce a design that physically fit together and mounted the components as intended, however when combined with the software and electronics, I encountered a few problems that I had to overcome by making some design decisions.

4 Project Plan

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

5 Evaluation Plan

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

6 Safety Plan

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

7 References

References

[1] RS Components. Rs pro 18 awg hook up wire, pvc insulation. `https://docs.rs-online.com/770d/A700000007035534.pdf`, . Accessed: 22-01-2024.

[2] RS Components. Rs pro c14 snap-in iec connector. `https://docs.rs-online.com/c6ad/0900766b815867b2.pdf`, . Accessed: 22-01-2024.

[3] DFRobot. 7 inch hdmi display with usb touchscreen. https://www.farnell.com/datasheets/3162025.pdf. Accessed: 19-01-2024.

[4] Amol I. Dhenge, Nāgpur, and A. S. Khobragade. Mechanical nut-bolt sorting using principle component analysis and artificial neural network. 2013. URL https://api.semanticscholar.org/CorpusID:201742258.

[5] Raspberry Pi Foundation. Raspberry pi 4 model b. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/. Accessed: 19-01-2024.

[6] FreeCAD. Freecad. https://www.freecadweb.org/. Accessed: 19-01-2024.

[7] Infineon. Irlz44n datasheet. https://www.infineon.com/dgdl/Infineon-IRLZ44N-DataSheet-v01_01-EN.pdf?fileId=5546d462533600a40153567217c32725. Accessed: 22-01-2024.

[8] Katranji. Xl4015 datasheet. https://www.katranji.com/tocimages/files/457120-274359.pdf. Accessed: 22-01-2024.

[9] MyreMylar. Pygame gui. https://github.com/MyreMylar/pygame_gui/tree/0cbf7056518377b455d51a8d20167f4029756ad9. Accessed: 22-01-2024.

[10] Le Nam, Nguyen Mui, and Dang Tu. A method to design vibratory bowl feeder by using fem modal analysis. *Vietnam Journal of Science and Technology*, 57:102, 02 2019. doi: 10.15625/2525-2518/57/1/12859.

[11] Okdo. Okdo, camera module, csi-2 with 2592 x 1944 resolution. https://uk.rs-online.com/web/p/raspberry-pi-cameras/2020456/, . Accessed: 19-01-2024.

[12] Okdo. Okdo, camera module, csi-2 with 2592 x 1944 resolution specification. https://docs.rs-online.com/b064/A700000006917308.pdf, . Accessed: 19-01-2024.

[13] Pygame. Pygame documentation. https://www.pygame.org/docs/. Accessed: 22-01-2024.

[14] Pylint. Pylint. https://github.com/pylint-dev/pylint. Accessed: 19-01-2024.

[15] Python. Python. https://www.python.org/. Accessed: 19-01-2024.

[16] Gunther Reinhart and Michael Loy. Design of a modular feeder for optimal operating performance. *CIRP Journal of Manufacturing Science and Technology*, 3(3):191–195, 2010. ISSN 1755-5817. doi: https://doi.org/10.1016/j.cirpj.2010.09.003. URL https://www.sciencedirect.com/science/article/pii/S1755581710000908.

[17] Richard Silversides, Jian S Dai, and Lakmal Seneviratne. Force Analysis of a Vibratory Bowl Feeder for Automatic Assembly. *Journal of Mechanical Design*, 127(4):637–645, 08 2004. ISSN 1050-0472. doi: 10.1115/1.1897407. URL https://doi.org/10.1115/1.1897407.

[18] Tkinter. Tkinter documentation. https://docs.python.org/3/library/tkinter.html. Accessed: 22-01-2024.

[19] Z. Zhang and Massachusetts Institute of Technology. Department of Mechanical Engineering. *Design and Development of an Automated Sorting and Orienting Machine for Vials*. Massachusetts Institute of Technology, Department of Mechanical Engineering, 2019. URL https://books.google.co.uk/books?id=Mi5WzQEACAAJ.