Final Design Report

Chilli Powder Corp (CPC)

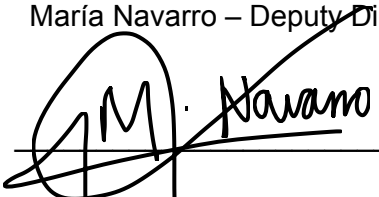
“Pepper-tual Hardness Tester”

Final Design Report **13/02/24**

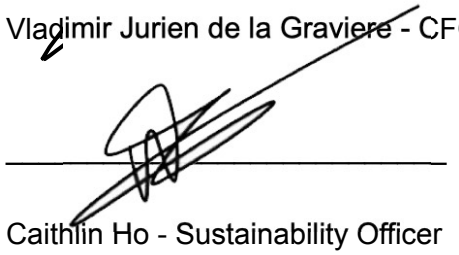
Approved by

Dámaso Matheus - Director

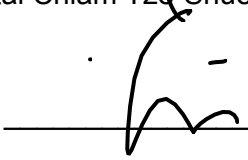
María Navarro – Deputy Director



Vladimir Jurien de la Graviere - CFO



Petal Chiam Tze-Shuen – EDI Officer



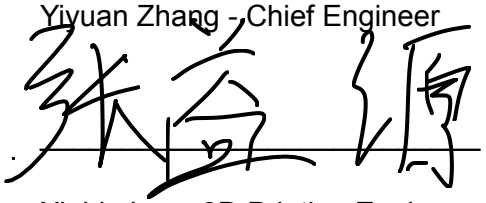
Caithlin Ho - Sustainability Officer

Catherine Saat - Chief of Automation and Control

Adrian Ng - Chief of Production

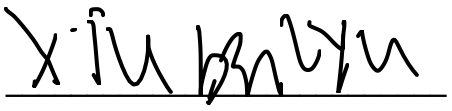
Dave Currie - Deputy Chief Engineer

Yiyuan Zhang - Chief Engineer



Dai Al-Mufti - Laser Cutter Engineer

Xiubin Lyu - 3D Printing Engineer



Contents

Project Summary

Chilli Powder Corp (CPC) was tasked to design and create a machine to test the hardness of various compacted powders. The final design is constructed to be automated as much as possible. This means that the user simply needs to push a button and recycle the tested powder themselves. The rest - including the measurement of the powder to be tested - is automated on a timer.

Various designs of the machine were considered before manufacture, with each design's potential strong points as well as shortcomings taken into account. Ultimately, we decided on an automated design that is both feasible and accurate. Not only does our design allow a highly accurate hardness measurement, but it also allows easier mass testing, as there can be a large volume of powder added to the machine before the procedure starts. It is also easy to recycle tested powder, as it can be removed manually from the crucible - and simply returned to the loader - until sufficient repeats have been conducted. The size of the machine was designed to be compact and small for easier testing, transportation as well as ensuring sustainability of the machine design. The machine is divided into four key parts: loading, compression, indentation, and image analysis. The powder travels to each step in the crucible, which is moved around the machine by pushers. The general machine function is as follows:

First, an excess amount of powder is put in the loader by the user, and the loader funnels 20g of powder into the crucible. It is then pushed along the platform to the compression step, where the powder is compressed. It then moves to the indentation step, which works similarly to the compression step, and a small indent is made on the surface of the powder. It is then taken to the image analysis step, and a camera takes a picture of the indent and analyses it to calculate the hardness. The crucible can then be picked up by the user and the powder can be recycled back into the loader, where the cycle is repeated.

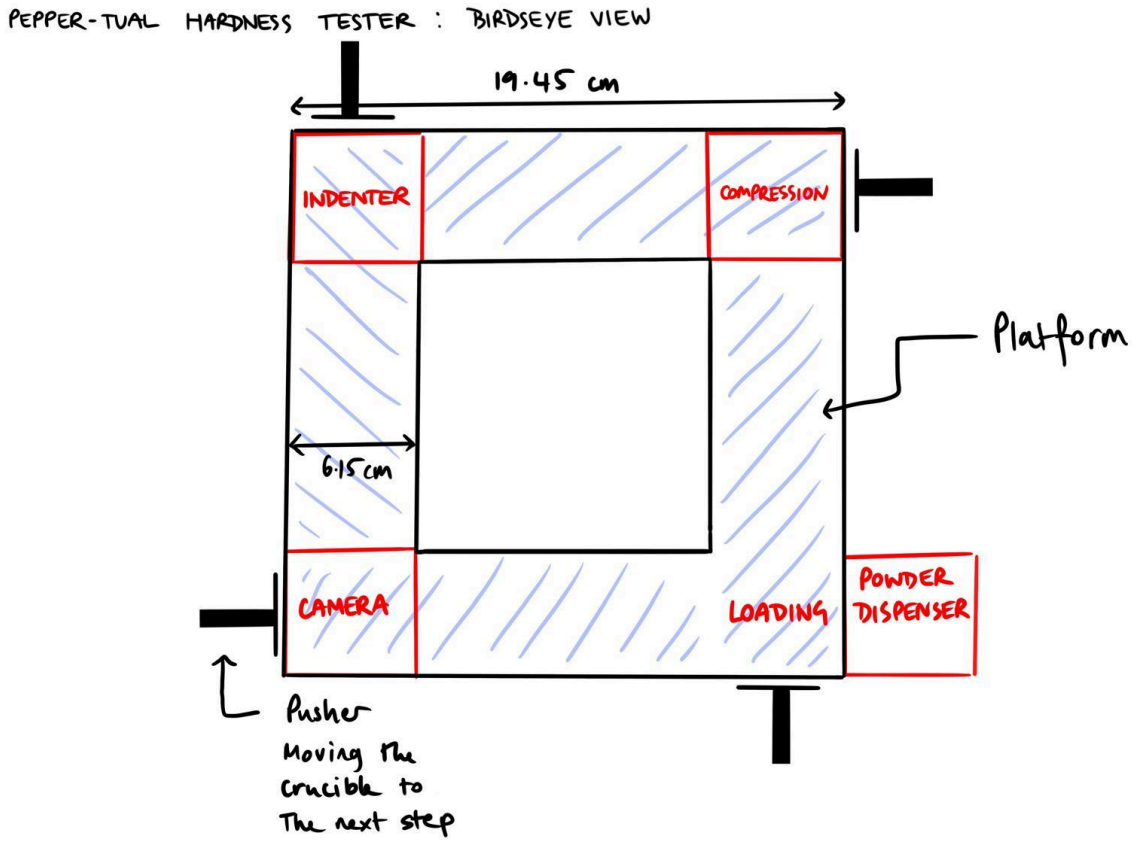


Figure 1: A birds-eye sketch of the machine

Even though the final version functions well, it was not without a few challenges and sacrifices that we were able to build it.

Initially, our machine design did not include any form of automation for the loading and unloading of the powder into the crucible. The original design was simply moving the crucible in a circular motion around the machine or moving different arms of the machine above the crucible to take turns to unload powder into it, compress, and indent the powder in the crucible. For these designs, we explored both circular and linear movements of the machine and ultimately decided on having the crucible follow a circular movement to ensure the same starting and ending position of the crucible as this increases efficiency of the automation process.

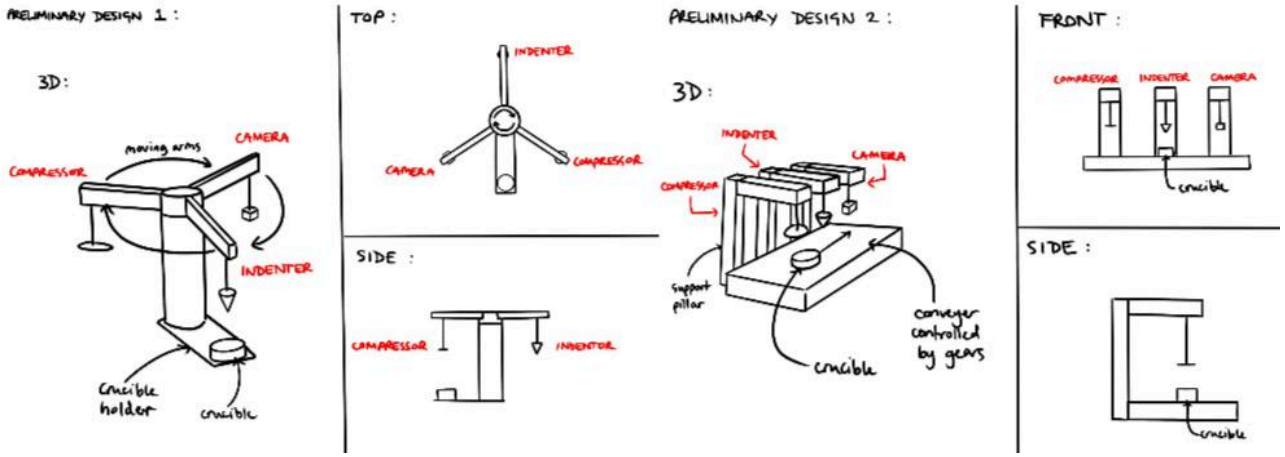


Figure 2: a) Initial machine design following a circular movement design b) Initial machine design following a linear movement design

As such, major revisions of the design were made at different stages of development. For example, in the concept design, the crucible was planned to be moved around to each stage by a conveyor belt rather than the pushers. We found that operating multiple conveyor belts on our machine would cost a lot of power and material, as well as be prone to more mechanical failures. We replaced the conveyor belts with a smooth PLA surface, which is more simple to manufacture and incorporate into the main body of the machine (refer to chapter 2.1.5 for more detail).

We also planned to make the machine fully automatic by including an ‘unloader’, which would have dumped the tested powder into a compartment attached to the machine. After careful planning and testing, we unfortunately are not able to implement it into the machine without significant changes to the design of the main body of the machine. It would also have been difficult to wire into the rest of the electronics without causing issues, such as a higher power requirement. Due to the loader, it is straightforward to conduct multiple tests on a specific powder in a short amount of time.

The following report details our final concept and operation of the machine, as well as outlines the sustainability of the machine, financial costs incurred and materials used.

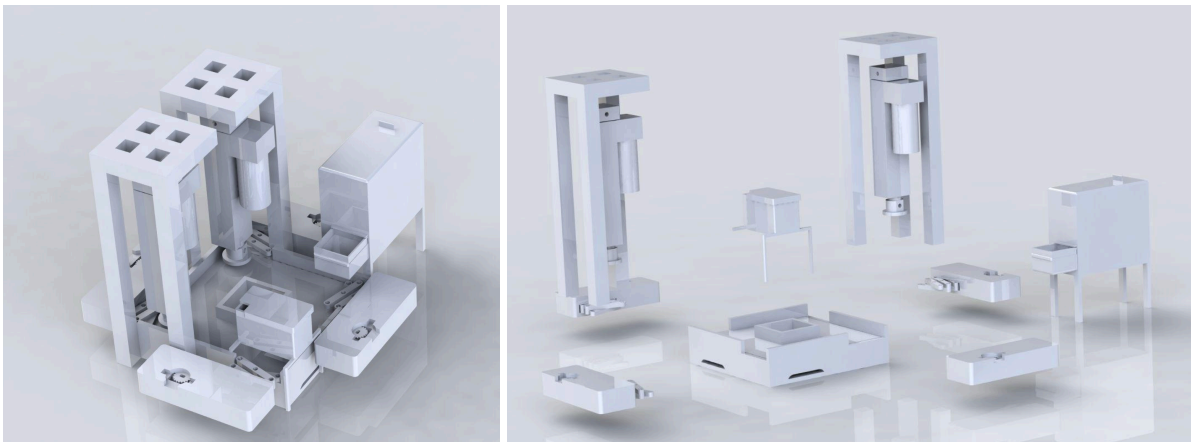


Figure 3: A 3D render of the full assembly of the machine (left) and an ‘exploded’ view of the machine (right)

Design Overview

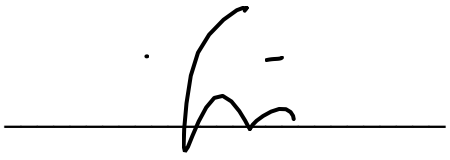
2.0 - User Operation

The instructions will be given as a list of instruction for an accurate experience:

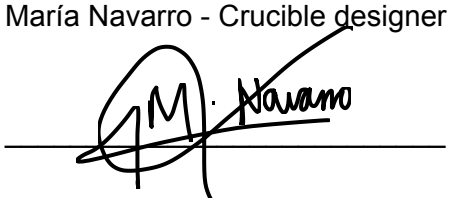
1. Plug the machine to a 230V outlet and turn on the switch to turn the machine on.
2. Wait 20 sec to make sure the powder is well dispensed.
3. The machine will begin the compressing process and move on to the indentation process.
4. Once the crucible has reached the image analysis, run the code on the prewired computer with a split screen setup (one half displaying a live stream, the other half showing the code).
5. The crucible will return to the initial corner where the powder dispensing took place. Swap out the crucible with a new one, ready for the next test.

Part 1 - Crucible, Platform and Movement

Petal Chiam Tze-Shuen - Platform designer

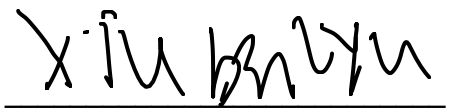


María Navarro - Crucible designer



Yiyuan Zhang - Platform designer

Xiubin Lyu - Crucible manufacturer



Catherine Saat - Pusher programmer

Dai Al-Mufti - Pusher designer and manufacturer

Dave Currie - Crucible designer

2.1.1 – Crucible

The crucible is a hollow cylinder with a base, and has an outer radius of 5.9 cm, a wall thickness of 0.85 cm, and a height of 1.7 cm. It is 3D printed at 100% infill to withstand the high pressure exerted by the actuators.

Given the small amount of powder used each time for testing, the crucible is designed such that the height of the powder relative to the crucible will only be slightly lesser. Since the indenter and compressor will perform its respective function at a specific timestamp, it is crucial that the crucible will only move in its predicted linear path than a random ‘zig-zag’ manner to arrive at each location at the specific time. As such, the diameter of the crucible is 98% of the width of the belt to ensure that the crucible will not have much room to move when pushed by the pushers.

Although the crucible was initially planned to be made of cast iron to facilitate the unloading step and meet the requirement of being partly conductive, this is no longer necessary as the unloading step has been removed.

2.1.2 - Platform

The platform is made of two main components: the track and the body. The track is the path that the crucible follows to each step, and the body acts as a central unit to unify all of the separate parts into one machine.

The track is made of PLA with 100% infill. It is designed to guide the crucible through each step while being able to endure the force of the compressor and indenter. Its low friction with the crucible reduces the load required by the pushers. The track measures 19 cm in length and width. It features a central space to accommodate the body's central walls, which measure 7.15 cm in length and width, along with the compressor and indenter stands.

The body, constructed from plywood, serves as the machine's skeleton. Inside, it houses the wiring, Arduino board, and supports for the track. Plywood is chosen for its ease of manufacture. It can be laser cut to precise dimensions, including small gaps in the outer walls for wiring connections between the motors and the Arduino board. Additionally, plywood is cost-effective and recyclable, making it an ideal material for this application.

2.1.3 - Movement

The machine's movement is fully automated and follows a precise sequence to determine the hardness of a powder material. The crucible is moved by four pushers, each guiding it to a different corner. The movement is stopped at each corner by a wall, ensuring accurate positioning for compressing, indentation, or image analysis. This process is automated, with the user only needing to run a code for image analysis. The crucible then returns to the starting point for replacement, ready for the next test.

Three of the pushers are linear actuators, and are controlled by the code shown in Figure 4. The final pusher only needs to move the crucible a small amount so that it can be retrieved by the user after image analysis.

The ‘controlPusher’ function enables the first three pushers to move the crucible between the loading step, compression step, indentation step and the image analysis step. The specific pins chosen offer pulse width modulation (PWM) control, which are needed in order to allow the actuators to function.

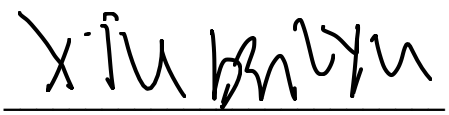
```
void controlPusher(int enaPin, int in1Pin, int in2Pin) //Controls the pushers
{
  // Extend the pusher
  digitalWrite(in1Pin, HIGH);
  digitalWrite(in2Pin, LOW);
  delay(5000); // Time taken for extension, adjust time as needed
  analogWrite(enaPin, 0); // Actuator stops extending
  delay(5000);
  analogWrite(enaPin, 255);

  // Retract the pusher
  digitalWrite(in1Pin, LOW);
  digitalWrite(in2Pin, HIGH);
  delay(5000); // Time taken for extension, adjust time as needed
  analogWrite(enaPin, 0); // Actuator stops retracting
  delay(5000);
  analogWrite(enaPin, 255);
}
```

Figure 4: The code that controls the pushers

Part 2 - Loading

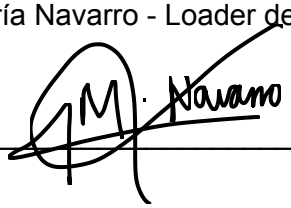
Dave Currie - Loader designer

Xiubin Lyu - Loader manufacturer



Catherine Saat - Loader programmer

María Navarro - Loader designer



The loading mechanism of our design is inspired by dog food dispensers, utilising a similar approach to precisely dispense 20g of powder. The main body of the dispenser is filled with powder, and within it, there is a platform connected to an open funnel above the crucible. The 'dogbowl' is a hollow rectangular container without a base. The platform moves at a rate of approximately 30 RPM. The dogbowl is programmed to move by the gears connected to the motor for 6 seconds until it is fully on the platform. During this time, powder from the main body is dispensed into the dogbowl for a programmed period of 5 seconds. Afterward, the gears rotate anticlockwise to return the dogbowl to its starting position.

Due to the absence of a base, the powder in the dogbowl naturally falls into the funnel and crucible below. The funnel provides an accurate pathway, ensuring the powder falls directly into the centre of the crucible, minimising potential spills. The fixed volume of the dogbowl allows for precise measurement, as the volume corresponding to 20g is determined based on the powder's density, ensuring accurate dispensing.

During the construction process, the 3D printer was not accurate and some of the parts did not fit well with each other, hence slight adjustments to the CAD files were made to compensate for the slight differences in dimensions when printed to resolve the issue.

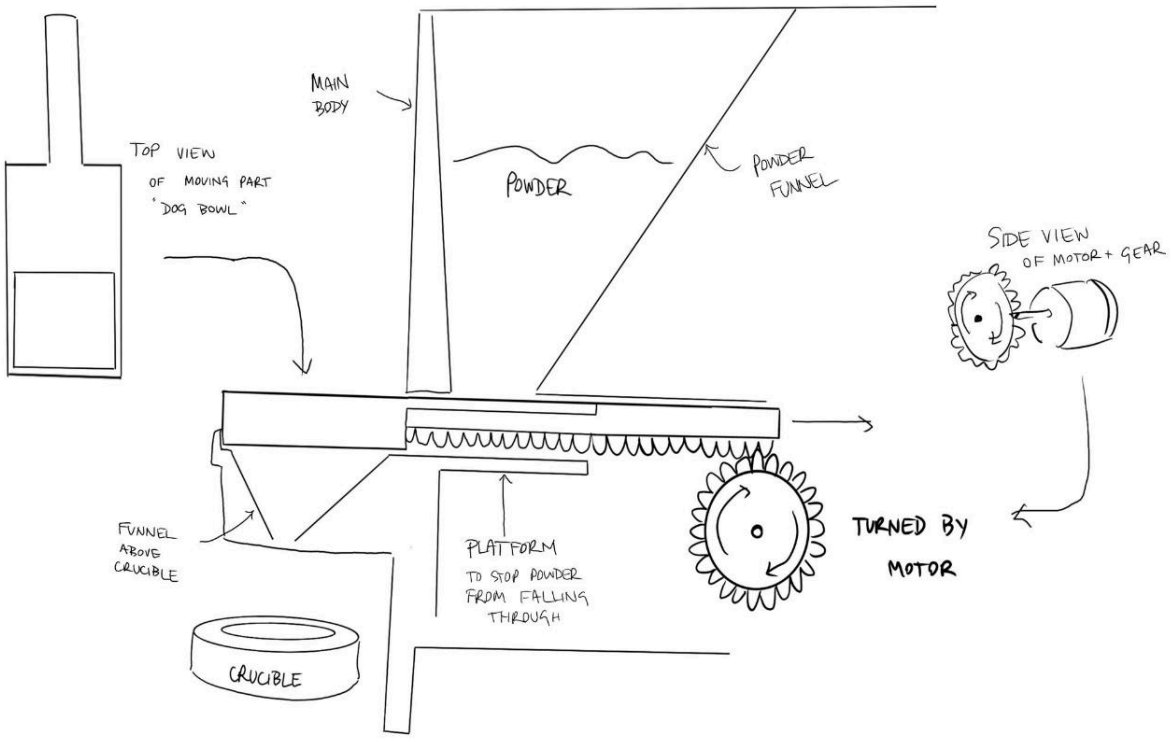


Figure 5: Magnified view of the loading component of powder into the crucible.

Figure 6: A 3D render of the loading mechanism

The code provided uses functions of timed delays and precise motor control functions to ensure that each step can be executed at a precise timestamp. Furthermore, the delay values are carefully chosen to ensure that the mechanical action at each stage is completed before the crucible is moved off to the next step.

For this phase, an excess amount of powder is added to the loader. The 'runStepperForTime' function is implemented to control the 28BYJ-48 stepper motor using the Stepper library in Arduino. The stepper motor is powered using a DC input plug of 12V. The steps per revolution for the stepper motor is set at 1024 to ensure a smoother operation of the loading of powder at 30 RPM. The stepper motor moves for a specified number of steps, waits for a set period of time, then reverses its direction. 20g of powder will then be funnelled into the crucible for the powder to be compacted.

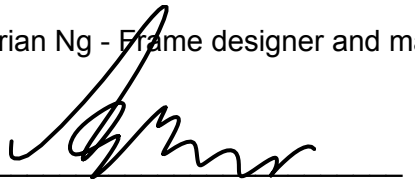
```
#include <Stepper.h>

// Stepper motor constants
const int stepsPerRevolution = 1024;  // Adjusts how smooth the operation of the stepper motor is
const int rolePerMinute = 30;  // Adjustable range of 28BYJ-48 stepper is 0~17 rpm

// Stepper pin numbers, adjust according to stepper motor location
Stepper stepperLoader(stepsPerRevolution, 29, 25, 27, 23); // Loader
Stepper stepperPusher(stepsPerRevolution, 33, 37, 35, 39); // Pusher 4

void runStepperForTime(Stepper & stepper, int steps)
```

```
// Runs the stepper for x seconds, stops for y seconds,
// Then runs the stepper in the opposite direction for n seconds, then stops for m seconds
// All timings can be defined in constants
{
  stepper.step(steps);
  delay(5000); //stops for 5 seconds
  stepper.step(-steps);
  delay(5000);
}
```

Figure 7: The code that controls the stepper motors

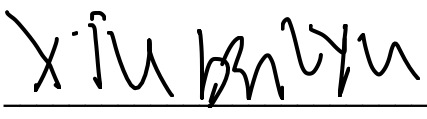
Part 3 - Compression and Indentation

Adrian Ng - Frame designer and manufacturer

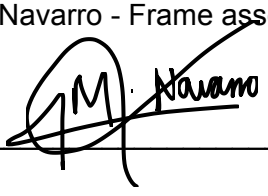


Catherine Saat - Actuator function

Xiubin Lyu - Actuator attachment manufacturer



María Navarro - Frame assembly



For compression, since the entire movement of the crucible on the platform of the machine is timed, the entire loading process would take 20 seconds, after which the crucible now containing the powder would have been pushed by the pushers such that it is positioned directly under the actuator. An arduino signal activates the extension of the actuator, causing the actuator to apply pressure onto the powder and compress it. The linear actuator has a fixed power rating of 20W. By keeping the time of force application constant, we are then able to ensure consistency in the force applied and hence consistency in the extent of compression on the powder achieved. The compressor is attached to a plate which is used to compress the powder in the crucible. This plate has the same diameter of the crucible to ensure even compression of powder throughout the crucible.

For indentation, the crucible is pushed by the pusher until it lies directly under the second linear actuator, which takes the same amount of time as the compression step. A 3d-printed head made of PLA with a tip diameter of 2mm attached to the indenter is pressed into the compacted powder.

When designing, we had to ensure that the frames of both actuators were strong enough to withstand the reaction force due to large force applied when compressing and indenting the powder in the crucible such that the frames hold onto the actuators firmly. To overcome this, frames must be made of materials with high tensile strength, thus we used aluminium frames screwed to the entire assembly.

Figure 8: A 3D render of the compressor mechanism

At this stage, the ‘controlActuator’ function uses the delay function to extend and retract the linear actuator to compress the powder in the crucible within a specified period of time. Similarly, after the powder is compacted, the crucible is pushed to the indentation stage using a similar ‘controlPusher’ function. Another actuator then indents the powder such that an image analysis can be performed via the Monte Carlo simulation. Finally, the crucible is moved back to its original position using a second stepper motor such that the crucible is replaced with a new one.

```
void controlActuator(int enaPin, int in1Pin, int in2Pin) //Controls the movement of the linear actuator (extends and retracts)
{
  // Extend the actuator
  digitalWrite(in1Pin, HIGH);
  digitalWrite(in2Pin, LOW);
  delay(2000); // Time taken for extension, adjust time as needed
  analogWrite(enaPin, 0); // Actuator stops extending
  delay(2000);
  analogWrite(enaPin, 255);

  // Retract the actuator
  digitalWrite(in1Pin, LOW);
  digitalWrite(in2Pin, HIGH);
  delay(2000); // Time taken for extension, adjust time as needed
  analogWrite(enaPin, 0); // Actuator stops retracting
  delay(2000);
  analogWrite(enaPin, 255);
}
```

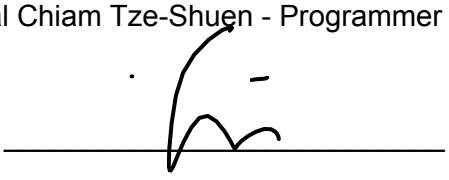
Figure 9: The code that controls the compressor and the indenter

Part 4 - Image Analysis

Dámaso Matheus - Programmer

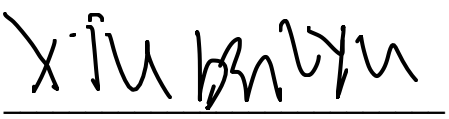
Catherine Saat - Programmer

Petal Chiam Tze-Shuen - Programmer



Dave Currie - Camera case designer

Xiubin Lyu - Camera case manufacturer



The image capture process uses a camera held above the third corner in a 3D printed case. At 50s, the crucible moves directly under the camera, and an arduino command based on the set timer takes a high-resolution picture of the indented sample to identify the size of the indent. The image is sent to an external computer to be analysed.

The analysis is performed through the utilisation of the cv2 library. The image used for analysis is found through finding the most recent image in the relevant folder. Then it is converted into greyscale, where then a threshold value is applied. This creates an image where the pixels below the threshold value will be black and the pixels above will be white, creating a completely black and white image, where the indentation shape will be a solid block of either black or white. This allows for the contour function to then create a contour line of the indentation, which allows for the area of the contour shape to be found. Finally the hardness is found by finding the percentage the indentation shape takes up in the overall image. The smaller the percentage of the indentation shape, the harder the powder.

Figure 10: A 3D render of the camera casing

Reports

3.1 – Sustainability Report

The machine is designed using responsibly sourced materials for different machine components. The main structural elements including walls and floors will be made from sustainably sourced wood, which is recyclable as it can be easily reused as smaller parts of other construction models, and if it must be discarded it is nonetheless biodegradable. Wood also has a smaller carbon footprint, as well as a lower energy and water life cycle compared to other materials.

More intricate parts of the machine - such as the indenter and compressor heads - will be 3D printed. All 3D printed components are made from polylactic acid (PLA), due to being able to produce aesthetic precise results. PLA is made from biodegradable natural resources as well. Additionally, our machine is small, minimising the total possible waste generated in the unlikely event where the entire machine has to be discarded.

For the computing aspect, we will use an Arduino board to control the various machine functions such as the compressor and conveyor belt. This board is land-fill safe, ensuring minimal environmental impact if disposal becomes necessary, although both the board and accompanying wires are meant to be recycled for future projects or electrical designs, again minimising waste.

In order to save power and reduce our use of unrecyclable materials, we will power our machine using the mains power supply rather than a battery.

3.2 – EDI Report

A final survey was conducted among team members to gather feedback regarding our collaboration throughout the course of the project. While the team faced some challenges initially due to uneven gender distribution, language barriers and fear to share their ideas, we have put in place certain measures to address these issues to foster a more inclusive and effective team collaboration. We scheduled regular check-in sessions with the members of the team to understand their concerns and address them. When working in separate teams, we allocated tasks to each individual based on their likings and strong suits. When a member faces a technical problem, we will gather as a team to problem solve and come up with alternative solutions to ensure each member does not struggle on his or her own. Furthermore, we made sure to have clear and transparent communication channels where all project decisions and updates are documented and easily accessible. Ultimately, the responses (shown in Figure 11) collected were overwhelmingly positive. The majority of team members displayed satisfaction with team dynamics and project progress.

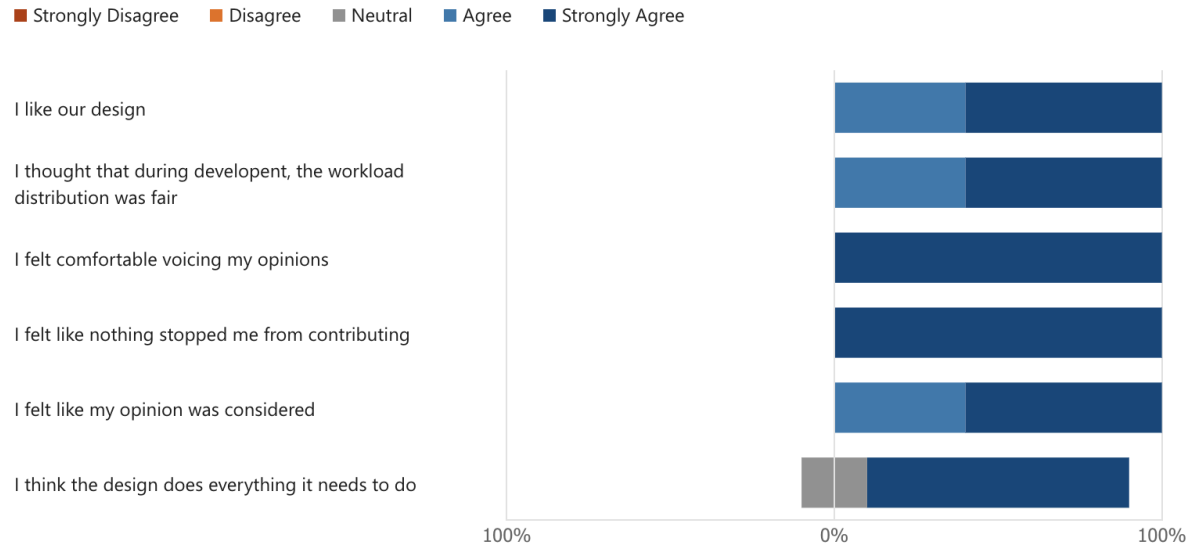


Figure 11: The results of the survey sent out to the whole team

Most felt confident expressing their opinions, as well as valued and recognised for their contributions. The survey assessed design satisfaction, comfort in sharing opinions, perceived barriers to contribution, workload management, and task clarity. Across these areas, nearly everyone agreed or strongly agreed, reflecting a collective confidence in our project's direction and execution.

Over the course of the project, each of us have displayed improvements in celebrating and acknowledging the diverse backgrounds of each team member. Our team is committed to equality, diversity, and inclusion. We value diverse perspectives and ensure every member feels valued. With the respective measures that we have put in place, we are proud that every member in our team feels a deep sense of team satisfaction in all aspects.

3.3 – Financial Report

Below is a summary of the ordered items for the machine, with a total expenditure of £198.91 and a gain of £7, leaving £8.09 from the original budget. The budget surplus we achieved demonstrates our careful financial planning and commitment to the project. Each expenditure was thoughtfully made, allowing for sufficient money to address any unforeseen issues.

The most significant expenditure was on the linear actuators, costing £119.75. We purchased five linear actuators: two at £28.90 each for indentation and compressing, and three to push the crucible. These actuators are necessary for providing sufficient force to move the crucible, compress and indent the powder.

Additionally, we bought two cameras: one at £7.99 which was not compatible with arduino, therefore we sold it on Gumtree for £7.00. The new camera was purchased at £16.99 and successfully captures an accurate image.

In order to lower our budget, we used high-quality 3D printers provided by the department to manufacture the majority of the machine's parts. Initially, we fabricated four pushers in-house, but they lacked the precision needed to push the crucible to the exact locations at the ends of the platform. We replaced three of these pushers with linear actuators, as they need to extend and retract 15 cm to push the crucible to the next step. The last pusher, which only requires a 5 cm extension from the image analysis stage to the manual unloading stage, was successfully fabricated in-house with the necessary precision. We manually reposition the empty crucible under the loading step to restart the process. For the remaining items, we sourced them from multiple websites to ensure the lowest prices possible.

Overall, our budget management was efficient, allowing us to accommodate any further expenses after the testing phase while still staying well within the budget.

Name	Link	Unity Price	Quantity	Full Spendings	Left to Spend
Linear actuator	https://www.amazon.co	28.9	2	57.8	
Camera n°1	https://www.amazon.co	7.99	1	7.99	
Ball Bearing	https://www.amazon.co	0.6	10	6	
Plywood	https://hobarts.com/pro	9.47	3	28.41	
Power adaptator	https://www.amazon.co	6.59	3	19.77	
Linear actuator 150mm	https://www.amazon.co	18.03	2	36.06	
Linear actuator	https://www.amazon.co	25.89	1	25.89	
Camera n°2	https://www.amazon.co	16.99	1	16.99	
Sold item		7	1		7
Total				191.91	8.09

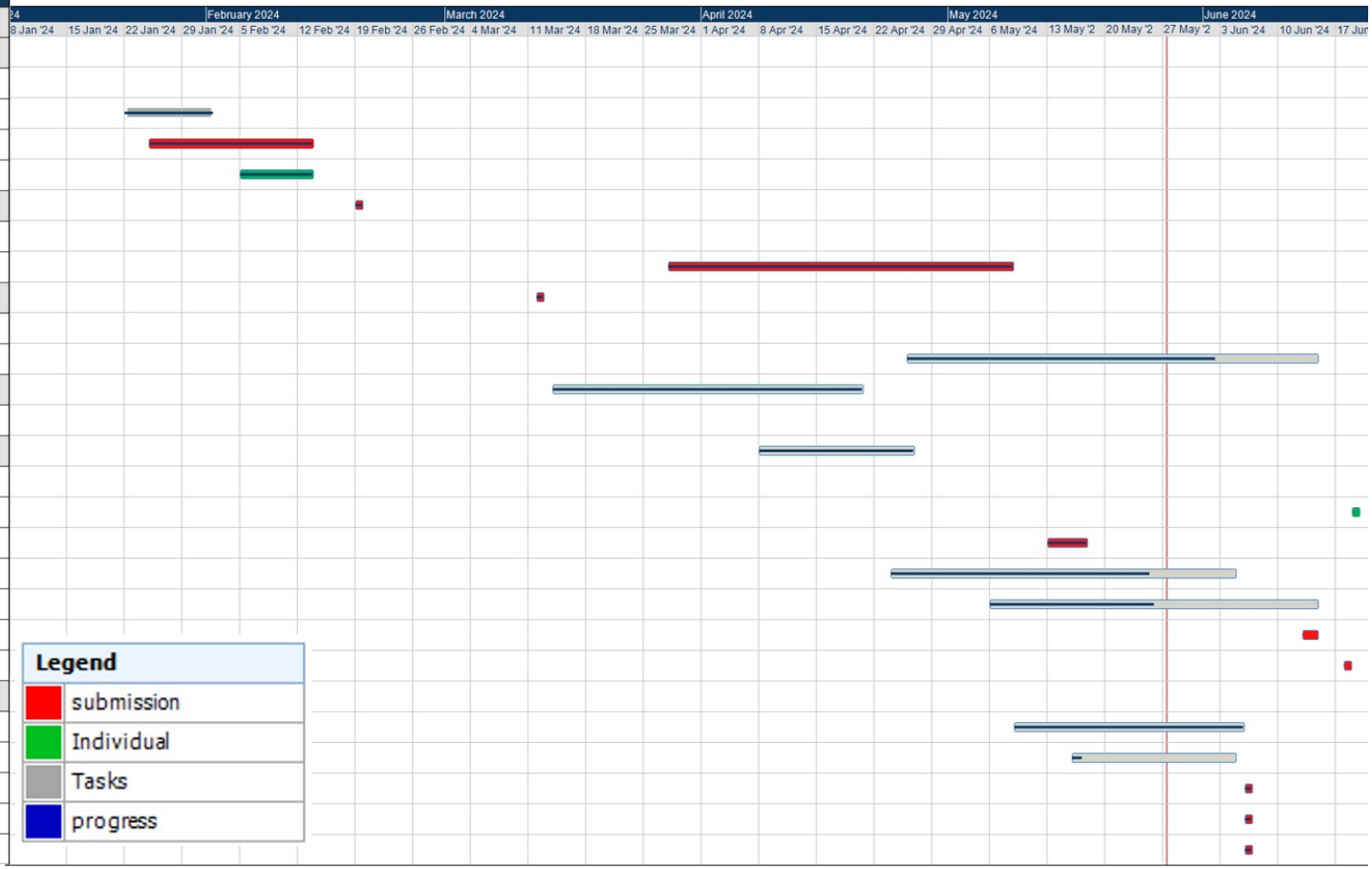
Item on ebay/gumtree /facebook marketplace	Price	Sold
Arduino Camera	7	Yes
Ball Bearings	2	Pending
Total		7

3.4 – Risk Assessment Report

Hazard	Person at Risk	Existing Precautions	Extra Precautions	Emergency Actions
Electrocution from wires	All members	Arduino boards have low voltage	Don't touch when machine in operation	Use insulating tape to cover wires
Pinching your hand	All members	Not specified	Don't touch when machine in operation	First aid kits available
Crushing your hand	All members	Not specified	Don't touch when machine in operation	First aid kits available
Trip hazard	All members	Not specified	Cables will be taped and secured	Not specified
Fire	All members	Fire extinguishers available in the building	Installing a fan to prevent machine components	Use fire extinguishers

3.5 – Project Timeline

	Task Name	Start	End	Priority
1	**Chili Powder Corp**	**15/01/2024**	**18/06/2024**	**100**
2	**Concept phase**	**15/01/2024**	**19/02/2024**	**30**
3	Schematic drawings	22/01/2024	01/02/2024	0
4	concept design report (15%)	25/01/2024	13/02/2024	15
5	Individual video submissions (5%)	05/02/2024	13/02/2024	5
6	Presentation of concept design (10%)	19/02/2024	19/02/2024	10
7	**Prototype phase**	**19/02/2024**	**26/04/2024**	**5**
8	Letter submission (5%)	28/03/2024	08/05/2024	5
9	Prototype test day (Pass/Fail)	12/03/2024	12/03/2024	0
10	**Manufacturing**	**19/02/2024**	**16/04/2024**	**0**
11	Laser cutting	14/03/2024	19/04/2024	0
12	3D printing	14/03/2024	20/04/2024	0
13	**Solidworks**	**15/03/2024**	**24/04/2024**	**0**
14	Technical drawing	08/04/2024	26/04/2024	0
15	**Final Design phase**	**24/04/2024**	**19/06/2024**	**30**
16	Peer marking contribution	19/06/2024	19/06/2024	500
17	3D recutting	24/04/2024	31/05/2024	500
18	Adjustment of technical drawing (5%)	13/05/2024	13/05/2024	5
19	3D reprinting	24/04/2024	04/06/2024	0
20	Integration of software	24/04/2024	04/06/2024	0
21	Small group design defense (internal asse.	13/06/2024	14/06/2024	15
22	Final Design Presentation 10%	18/06/2024	18/06/2024	10
23	**Final design Submission (30%)**	**08/05/2024**	**06/06/2024**	**35**
24	Final Report	08/05/2024	04/06/2024	0
25	Parts list	16/05/2024	04/06/2024	0
26	Production record	06/06/2024	06/06/2024	0
27	Technical drawings	06/06/2024	06/06/2024	0
28	Individual Video Submission 5%	06/06/2024	06/06/2024	5



Project Assessment

Overall, the final product is near-complete, with a few more things to be perfected. The machine code is fully functional - the timings of when each step should execute have been tested to be perfect, and the order of operations has also been completed.

Firstly, the various components of the machine need to be permanently attached to the body, thus completing the full assembly of the machine. This will be done with screws to ensure stability.

Some parts were not manufactured perfectly on the first try. For example, the crucible needed to be reprinted, as its diameter was made to be the exact same as the track width. This didn't account for manufacturing uncertainties, and the crucible was slightly too big to fit in the track.

The loader was also fully reprinted, as the supports were revised after some design revisions to the main body.

The pushers have also been completely overhauled. Initially, four pushers in total were planned to be manufactured. One has been fully assembled, and was tested before attempting to make the remaining pushers. However, the design has partially failed in testing, as currently it is only able to move the crucible short distances - it cannot extend to the required 15cm. As a consequence, the remaining three pushers have been replaced by linear actuators, and the manufactured pusher has been slightly modified such that it can push the crucible from underneath the camera - just far enough so that the user can retrieve it.

The required extension time of the compressor has yet to be tested. This must be adjusted, as if the actuator doesn't extend for enough time, the results will be inaccurate. If it extends too much, it could cause damage to the machine. Because of this, the extension time will start as an underestimate, and will be continuously adjusted upwards until the desired result is accomplished.

The camera has been tested successfully, and the image captured can successfully be sent to the connected computer ready for analysis. The image analysis code has also been successfully tested on sample images.

Once complete, not only will it satisfy the basic requirements of compression, indentation and image analysis, but it also allows easy and semi-automatic test repeats for large data sets.

Appendix

5.1 - Notes

In order to streamline the machine design and manufacture, the development of each step was allocated to small teams, rather than three subgroups.

Yiyuan Zhang was tasked to develop the unloading step, which involved designing the unloader itself, how it would operate and how it would integrate into the rest of the machine. Zhang was essential in developing this step. Unfortunately, due to time constraints, the unloading step was scrapped - and much of Zhang's efforts are not reflected in the final machine and therefore in the design overview of the final report.

Also, the order of signatures on the design overview cover pages are done in no particular order - just because an individual has their name on the top of the page, it doesn't mean they contributed the most to that subproject.

5.2 - Parts and production list

Parts LIst									
Part Category	Part No.	Description	Designed	Date of Final Version	Used?	Designer	Checked by	CAD File	Material
Loading	C7-S1-001	Lid of Loader	Y	27/2/24	Y	Dave	Damaso	Dispenser-TopCover.S	PLA
	C7-S1-002	Side of Loader	Y	27/2/24	Y	Dave	Dai	Dispenser-SideCover.S	PLA
	C7-S1-003	Powder Container	Y	27/2/24	Y	Dave	Dave	Dispenser-MainBody.S	PLA
	C7-S1-004	Loader Support	Y	27/2/24	Y	Dave	Adrian	Dispenser-Legs.SLDP	PLA
	C7-S1-005	Powder Funnel	Y	27/2/24	Y	Dave	Maria	DIspenser-DogBowl.SI	PLA
		Part Drawing	Y	6/5/24	Y	Dave	Adrian	DispenserDrawing.SLI	PLA
Compression	C7-S2-001	Compression attachme	Y	10/5/24	Y	Adrian	Dai	Compression attachme	PLA
	C7-S2-002	Frame attacher	Y	9/5/24	Y	Dai	Adrain	frame attacher part 1.S	PLA
	C7-S2-003	Compressor stand	Y	10/5/24	Y	Dai	Dave	final stand.SLDPRT	Aluminium
	C7-S2-004	Linear actuator	Y	29/4/24	Y	Dai	Adrian	Linear Actuator.SLDPF	Model
		Comporessor Assembly	Y	11/5/24	Y	Dai	Adrian	compressor assembly final.SLDASM	
Indentation	C7-S3-001	Indentor attachment	Y	10/5/24	Y	Adrian	Dai	indentation attachment	PLA
	C7-S3-002	Frame attacher	Y	9/5/24	Y	Adrian	Dai	frame part 2.SLDPRT	PLA
	C7-S3-003	Indentor stand	Y	10/5/24	Y	Adrian	Dai	final stand.SLDPRT	Alumium
	C7-S3-004	Linear actuator	Y	29/4/24	Y	Adrian	Dai	Linear Actuator.SLDPF	Model
		Indentor Assembly	Y	11/5/24	Y	Adrian	Dai	indentation assembly.SLDASM	
Crucible	C7-SC-001	Crucible	Y	13/5/24	Y	Dave	Maria	Cruible.SLDPRT	PLA
Camera	C7-S4-001	Camera Basin	Y	13/5/24	Y	Dave	Damaso	camera-holder.SLDPR	PLA
	C7-S4-002	Camera Support	Y	13/5/24	Y	Dave	Damaso	Camera-Legs.SLDPRT	PLA
	C7-S4-003	Basin Lid	Y	13/5/24	Y	Dave	Damaso	Camer-lid.SLDPRT	PLA
Pusher	C7-SP-001	Gear	Y	5/3/24	Y	Dai	Damaso	gearr.SLDPRT	Plywood
	C7-SP-002	Baseplate	Y	6/3/24	Y	Dai	Damaso	I like this cute baseplat	Plywood
	C7-SP-003	Support for rack and g	Y	7/3/24	Y	Dai	Damaso	layer 2.SLDPRT	Plywood
	C7-SP-004	Extender support	Y	9/3/24	Y	Dai	Damaso	layer 3.2.SLDPRT	Plywood
	C7-SP-005	Gap for extender	Y	11/3/24	Y	Dai	Damaso	layer 4.SLDPRT	Plywood
	C7-SP-006	Cover	Y	12/3/24	Y	Dai	Damaso	layer 5.SLDPRT	Plywood
	C7-SP-007	Rack	Y	13/3/24	Y	Dai	Damaso	more shapy.SLDPRT	Plywood
	C7-SP-008	extender	Y	14/3/24	Y	Dai	Damaso	scisor.SLDPRT	Plywood
	C7-SP-009	rack	Y	15/3/24	N	Dai	Damaso	shapy the rack.SLDPR	Plywood
		Part Assembly	Y	16/3/24	Y	Dai	Damaso	WAllet.SLDASM	Plywood
Unloading	C7-SU-001	A foldbale unloader	Y	23/3/24	N	Alex	Vladimir	new unloading.sldrp	PLA
Belt	C7-SB-001	Conveyor belt	N	Not designed	N	None	Vladimir	Nofile as not designed	NONE

Production Record									
Part Category	Part No.	CAD File	Date	Start	Finish	Settings	Success	Material	Checked by
Loading	C7-S1-001	Dispenser-TopCover.SLDPRT	3/3/24	09:00	16:00		Y	PLA	Alex
	C7-S1-002	Dispenser-SideCover.SLDPRT	3/3/24	10:00	17:00		Y	PLA	Alex
	C7-S1-003	Dispenser-MainBody.SLDPRT	3/3/24	09:30	17:30		N	PLA	
	C7-S1-003R1	Dispenser-MainBody.SLDPRT	4/3/24	09:30	17:30		Y	PLA	Alex
	C7-S1-004	Dispenser-Legs.SLDPRT	20/5/24	11:00	16:00		Y	PLA	Alex
	C7-S1-005	DIspenser-DogBowl.SLDPRT	3/3/24	10:00	00:00		Y	PLA	Alex
Compression	C7-S2-001	Compression attachment part 2.SLDPRT	20/5/24	09:00	16:00		Y	PLA	Dave
	C7-S2-002	frame attacher part 1.SLDPRT	20/5/24	10:00	17:00		Y	PLA	Dave
	C7-S2-003	final stand.SLDPRT	20/5/24	09:30	17:30		Y	Aluminium	Dave
	C7-S2-004	Linear Actuator.SLDPRT	29/4/24	11:00	16:00		Y	Model	Dave
Indentation	C7-S3-001	indentation attachment part 3.0 1.SLDPRT	20/5/24	09:00	16:00		Y	PLA	Dave
	C7-S3-002	frame part 2.SLDPRT	20/5/24	10:00	17:00		Y	PLA	Dave
	C7-S3-003	final stand.SLDPRT	20/5/24	09:30	17:30		Y	Alumium	Dave
	C7-S3-004	Linear Actuator.SLDPRT	29/4/24	11:00	16:00		Y	Model	Dave
Crucible	C7-SC-001	Cruible.SLDPRT	2/5/24	09:00	16:00		N	PLA	
	C7-SC-001R1	Cruible.SLDPRT	3/5/24	09:00	16:00		Y	PLA	Xiu Bin
Camera	C7-S4-001	camera-holder.SLDPRT	14/5/24	10:00	17:00		Y	PLA	Xiu Bin
	C7-S4-002	Camera-Legs.SLDPRT	14/5/24	09:30	17:30		Y	PLA	Xiu Bin
	C7-S4-003	Camer-lid.SLDPRT	14/5/24	11:00	16:00		Y	PLA	Xiu Bin
Pusher	C7-SP-001	gearr.SLDPRT	26/5/24	09:00	16:00		Y	Plywood	Adrian
	C7-SP-002	I like this cute baseplate.SLDPRT	26/5/24	10:00	17:00		Y	Plywood	Adrian
	C7-SP-003	layer 2.SLDPRT	27/5/24	09:30	17:30		N	Plywood	
	C7-SP-003R1	layer 2.SLDPRT	28/5/24	10:30	18:30		Y	Plywood	Adrian
	C7-SP-004	layer 3.2.SLDPRT	27/5/24	11:00	16:00		Y	Plywood	Adrian
	C7-SP-005	layer 4.SLDPRT	27/5/24	10:00	00:00		Y	Plywood	Adrian
	C7-SP-006	layer 5.SLDPRT	27/5/24	09:00	16:00		Y	Plywood	Adrian
	C7-SP-007	more shapy.SLDPRT	27/5/24	10:00	17:00		Y	Plywood	Adrian
	C7-SP-008	scisor.SLDPRT	27/5/24	09:30	17:30		Y	Plywood	Adrian
	C7-SP-009	shapy the rack.SLDPRT	27/5/24	11:00	16:00		Y	Plywood	Adrian
Belt	C7-SB-002	track.SLDPRT	21/5/24	14:00	00:00		Y	PLA	Adrian

5.3 - Arduino code

```
#include <Stepper.h>

// Stepper motor constants
const int stepsPerRevolution = 1024;  // Adjusts how smooth the operation of the stepper motor is
const int rolePerMinute = 30;  // Adjustable range of 28BYJ-48 stepper is 0~17 rpm

// Steps for 5 seconds of running
const int stepsForFiveSeconds = (rolePerMinute * stepsPerRevolution) / 12;
// Steps for 3 seconds of running
const int stepsForThreeSeconds = (rolePerMinute * stepsPerRevolution) / 20;

// Stepper pin numbers, adjust according to stepper motor location
Stepper stepperLoader(stepsPerRevolution, 29, 25, 27, 23); // Loader
Stepper stepperPusher(stepsPerRevolution, 33, 37, 35, 39); // Pusher 4

// Linear actuator constants and pins

//Actuators - enable and input pins to control the compression and indentation
```

```
const int ENA_PINC = 2; // Actuator 1 enable pin
const int IN1_PINC = 3; // Actuator 1 input pin 1
const int IN2_PINC = 4; // Actuator 1 input pin 2

const int ENB_PIND = 5; // Actuator 2 enable pin
const int IN3_PIND = 6; // Actuator 2 input pin 1
const int IN4_PIND = 7; // Actuator 2 input pin 2


//Pushers - enable and input pins to move the crucible between each stage
const int ENA_PINP1 = 8; // Pusher 1 enable pin
const int IN1_PINP1 = 9; // Pusher 1 input pin 1
const int IN2_PINP1 = 10; // Pusher 1 input pin 2

const int ENB_PINP2 = 11; // Pusher 2 enable pin
const int IN3_PINP2 = 12; // Pusher 2 input pin 1
const int IN4_PINP2 = 13; // Pusher 2 input pin 2

const int ENA_PINP3 = 44; // Pusher 3 enable pin
const int IN1_PINP3 = 45; // Pusher 3 input pin 1
const int IN2_PINP3 = 46; // Pusher 3 input pin 2

void setup() {
  // Initialize stepper motors, i.e set the speed of the stepper motors
  stepperLoader.setSpeed(rolePerMinute);
  stepperPusher.setSpeed(rolePerMinute);

  // Initialize digital pins as outputs for Actuator 1
  pinMode(ENA_PINC, OUTPUT);
  pinMode(IN1_PINC, OUTPUT);
  pinMode(IN2_PINC, OUTPUT);
  digitalWrite(ENA_PINC, HIGH);

  // Initialize digital pins as outputs for Actuator 2
  pinMode(ENB_PIND, OUTPUT);
  pinMode(IN3_PIND, OUTPUT);
  pinMode(IN4_PIND, OUTPUT);
  digitalWrite(ENB_PIND, HIGH);
```

```
  // Initialize digital pins as outputs for Pusher 1
  pinMode(ENA_PINP1, OUTPUT);
  pinMode(IN1_PINP1, OUTPUT);
  pinMode(IN2_PINP1, OUTPUT);
  digitalWrite(ENA_PINP1, HIGH);

  // Initialize digital pins as outputs for Pusher 2
  pinMode(ENB_PINP2, OUTPUT);
  pinMode(IN3_PINP2, OUTPUT);
  pinMode(IN4_PINP2, OUTPUT);
  digitalWrite(ENB_PINP2, HIGH);

  // Initialize digital pins as outputs for Pusher 3
  pinMode(ENA_PINP3, OUTPUT);
  pinMode(IN1_PINP3, OUTPUT);
  pinMode(IN2_PINP3, OUTPUT);
  digitalWrite(ENA_PINP3, HIGH);
}


void runStepperForTime(Stepper & stepper, int steps)
// Runs the stepper for x seconds, stops for y seconds,
// Then runs the stepper in the opposite direction for n seconds, then stops for m seconds
// All timings can be defined in constants
{
  stepper.step(steps);
  delay(5000); //stops for 5 seconds
  stepper.step(-steps);
  delay(5000);
}

void controlActuator(int enaPin, int in1Pin, int in2Pin) //Controls the movement of the linear
actuator (extends and retracts)
{
  // Extend the actuator
  digitalWrite(in1Pin, HIGH);
  digitalWrite(in2Pin, LOW);
  delay(2000); // Time taken for extension, adjust time as needed
  analogWrite(enaPin, 0); // Actuator stops extending
```

```
  delay(2000);
  analogWrite(enaPin, 255);

  // Retract the actuator
  digitalWrite(in1Pin, LOW);
  digitalWrite(in2Pin, HIGH);
  delay(2000); // Time taken for extension, adjust time as needed
  analogWrite(enaPin, 0); // Actuator stops retracting
  delay(2000);
  analogWrite(enaPin, 255);
}

void controlPusher(int enaPin, int in1Pin, int in2Pin) //Controls the pushers
{
  // Extend the pusher
  digitalWrite(in1Pin, HIGH);
  digitalWrite(in2Pin, LOW);
  delay(5000); // Time taken for extension, adjust time as needed
  analogWrite(enaPin, 0); // Actuator stops extending
  delay(5000);
  analogWrite(enaPin, 255);

  // Retract the pusher
  digitalWrite(in1Pin, LOW);
  digitalWrite(in2Pin, HIGH);
  delay(5000); // Time taken for extension, adjust time as needed
  analogWrite(enaPin, 0); // Actuator stops retracting
  delay(5000);
  analogWrite(enaPin, 255);
}


void loop() //The order of executed code
{
  runStepperForTime(stepperLoader, stepsForThreeSeconds); // loading
     controlPusher(ENA_PINP1, IN1_PINP1, IN2_PINP1); // pushes crucible towards
compressor

  controlActuator(ENA_PINC, IN1_PINC, IN2_PINC); // Actuator 1, which compresses
```

```
  delay(2000); // wait time to ensure actuator is out of way before pusher moves or if any issue arise during compression
  controlPusher(ENB_PINP2, IN3_PINP2, IN4_PINP2); //pushes crucible to indenter

  controlActuator(ENB_PIND, IN3_PIND, IN4_PIND); // Actuator 2, which indents
  delay(2000);// wait time in case there are any issues during indentation
  controlPusher(ENA_PINP3, IN1_PINP3, IN2_PINP3); // pushes crucible to camera

  runStepperForTime(stepperPusher, stepsForThreeSeconds); //pushes crucible to start
}
```

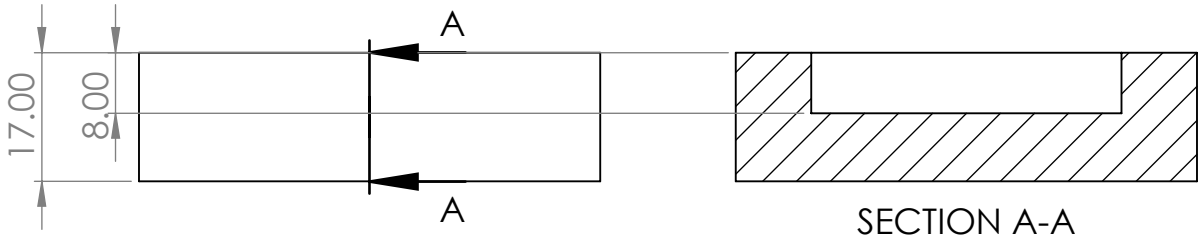
5.4 - Image analysis code

```
import cv2
import time

# Create a VideoCapture object for the webcam
cap = cv2.VideoCapture(1)  # 0 corresponds to the default webcam, you can change it if you have multiple cameras

# Function to generate a unique filename with a timestamp
def generate_filename():
    timestamp = time.strftime("%Y%m%d%H%M%S")
    return f'screenshot_{timestamp}.png'

# Read a single frame from the webcam
ret, frame = cap.read()

# Check if the frame is successfully captured
if ret:
    # Perform your analysis on the 'frame' here
    # You can use OpenCV functions or any other image processing libraries

    # For example, display the frame
    cv2.imshow('Webcam Feed', frame)

    # Generate a unique filename with a timestamp
    filename = generate_filename()

    # Save the screenshot with the unique filename
    cv2.imwrite(filename, frame)

# Release the VideoCapture and close all OpenCV windows
cap.release()
cv2.destroyAllWindows()
```

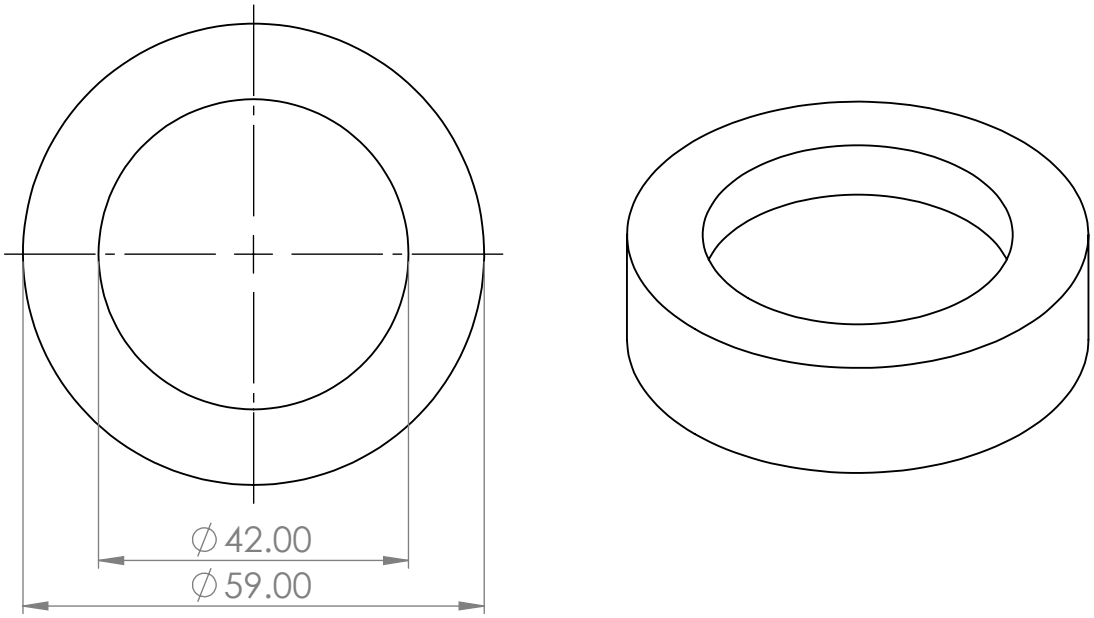
```
import cv2
import numpy as np
import os
import glob

# Function to find the most recent PNG file in a directory
def find_most_recent_png(directory):
    latest_file = None
    latest_timestamp = 0
    for file_name in glob.glob(os.path.join(directory, '*.png')):
        timestamp = os.path.getmtime(file_name)
        if timestamp > latest_timestamp:
            latest_timestamp = timestamp
            latest_file = file_name
    return latest_file

# Directory containing the PNG files
directory = 'C:/Users/catsa/Desktop/design study code/analysis_code' #change
accordingly

# Find the most recent PNG file
most_recent_image_path = find_most_recent_bmp(directory)

# Read the most recent image in grayscale
img = cv2.imread(most_recent_image_path, cv2.IMREAD_GRAYSCALE)

# Apply a threshold to create a binary image
threshold_value = 127
_, binary_image = cv2.threshold(img, threshold_value, 255, cv2.THRESH_BINARY)

# Display the binary image to ensure it has only black and white values
cv2.imshow("Binary Image", binary_image)
cv2.waitKey(0)
cv2.destroyAllWindows()

# Find the minimum pixel value in the image (the darkest value)
darkest_value = np.min(binary_image)

# Create a binary image where only the pixels with the darkest value are white
bw_image = np.where(binary_image == darkest_value, 255, 0).astype(np.uint8)

# Find contours in the binary image
contours, _ = cv2.findContours(bw_image, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

# Calculate the area for each contour
contour_areas = [cv2.contourArea(contour) for contour in contours]

# Calculate the overall image area
height, width = img.shape
overall_image_area = height * width

# Find the largest area and the percentage it occupies
if contour_areas:
```

```
    largest_area = max(contour_areas)
    percentage_of_image = (largest_area / overall_image_area) * 100

    print(f"Largest Contour Area: {largest_area:.2f}")
    print(f"Overall Image Area: {overall_image_area}")
    print(f"Percentage of Image Occupied by Largest Contour:
{percentage_of_image:.2f}%")
else:
    print("No contours found.")

# Draw the contours on a color version of the original image
img_with_contours = cv2.cvtColor(img, cv2.COLOR_GRAY2BGR)
cv2.drawContours(img_with_contours, contours, -1, (0, 255, 0), 2)  # Draw with green
color

# Display the image with contours drawn
cv2.imshow("Contours of Darkest Regions", img_with_contours)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

5.5 - Technical drawings

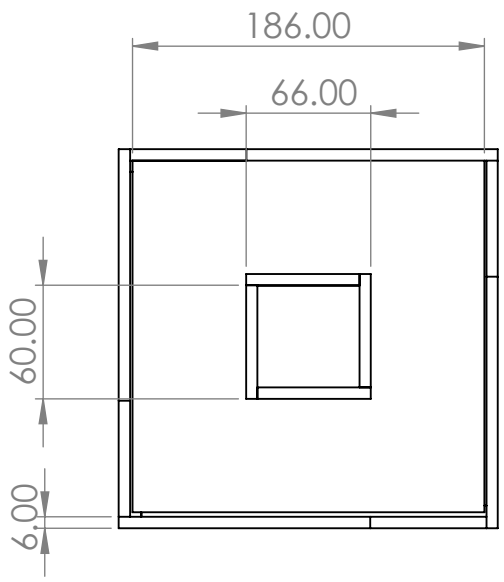


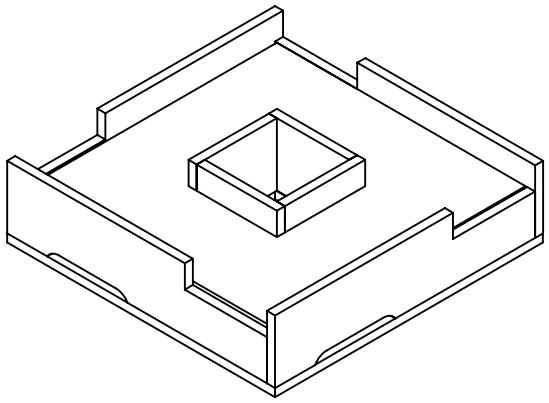
SECTION A-A
SCALE 1 : 1

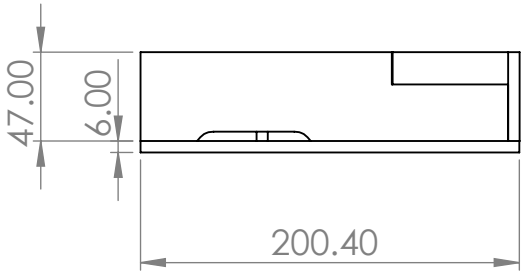


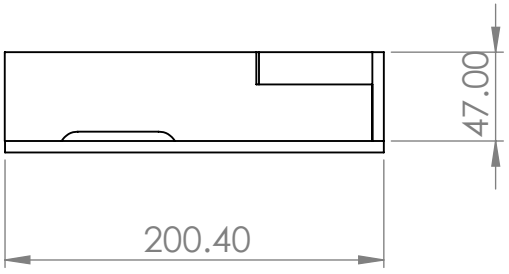
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: +- 0.5mm ANGULAR: +- 0.5 degrees		FINISH: 3D Printed		DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
	NAME	SIGNATURE	DATE			TITLE: Crucible	
DRAWN	CPC		6/5/2024				
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL: Red PLA		DWG NO. CPC-Crucible	A4
				WEIGHT:		SCALE:1:1	SHEET 1 OF 1

SOLIDWORKS Educational Product. For Instructional Use Only.









UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	3D Printed	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

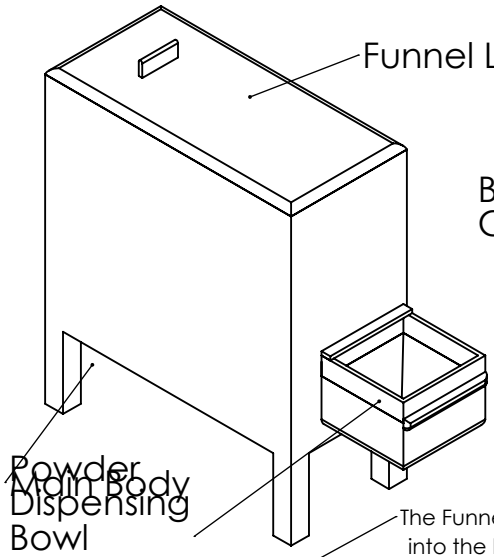
Platform

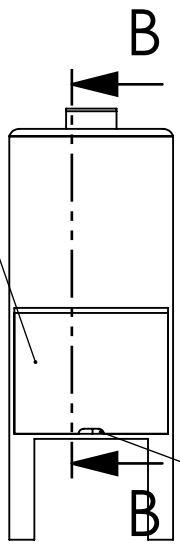
PLA + Wood

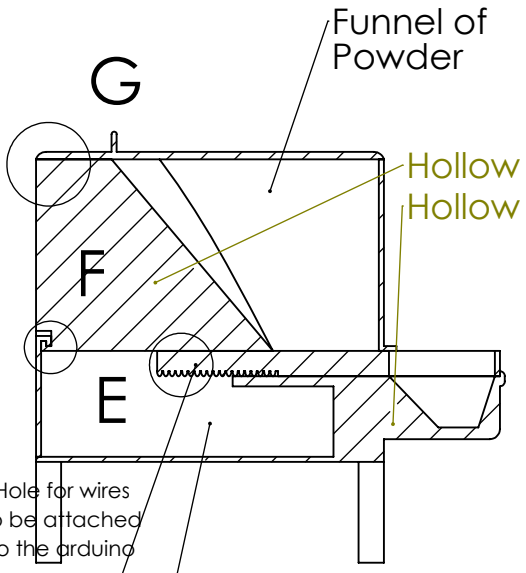
DWG NO. A4

WEIGHT: SCALE:1:5 SHEET 1 OF 1

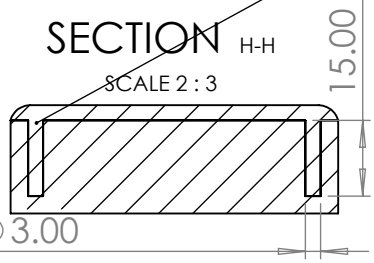
SOLIDWORKS Educational Product. For Instructional Use Only.

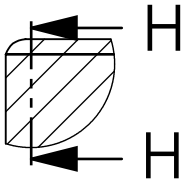




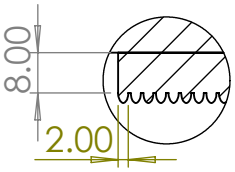


The Funnel Lid slots into the body via cylindrical pegs

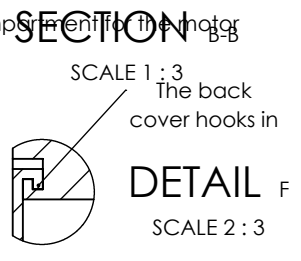


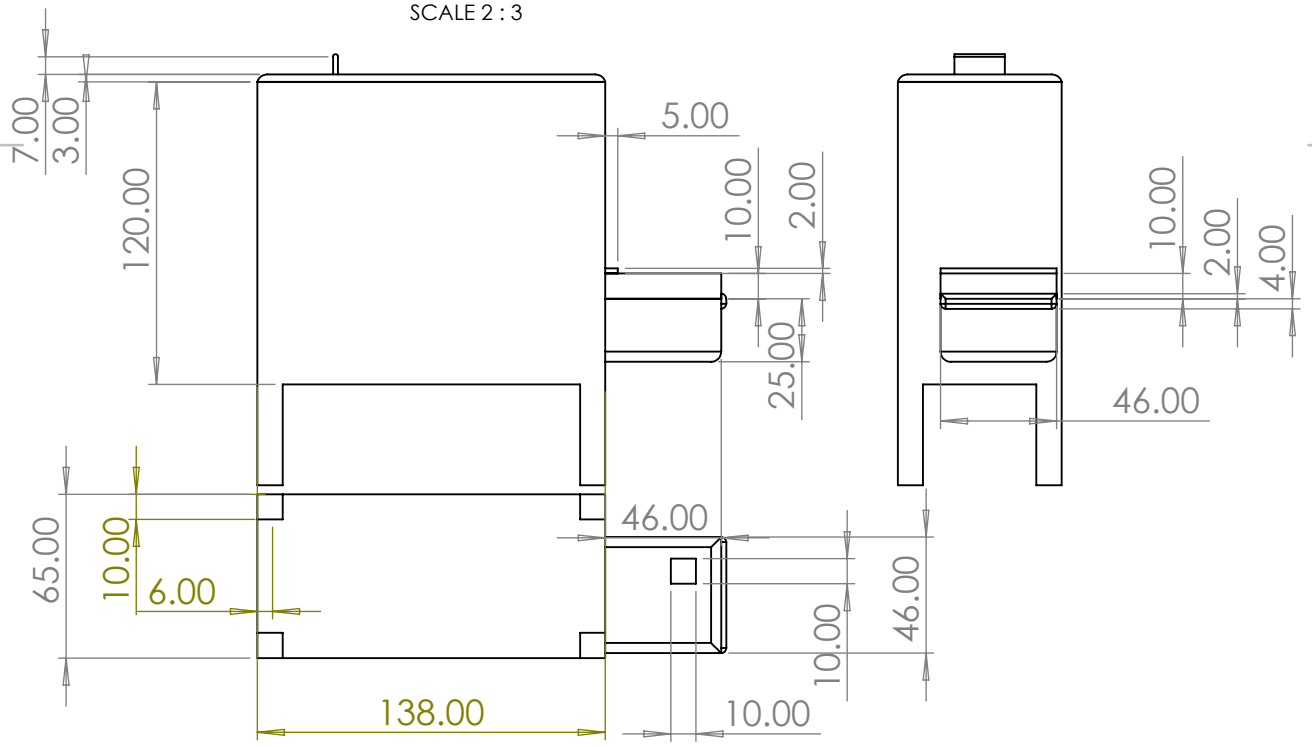


DETAIL G
SCALE 2 : 3



DETAIL E
SCALE 2 : 3

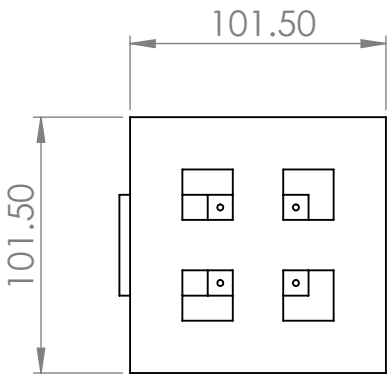


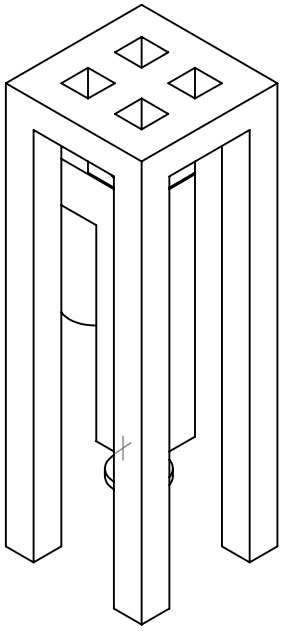


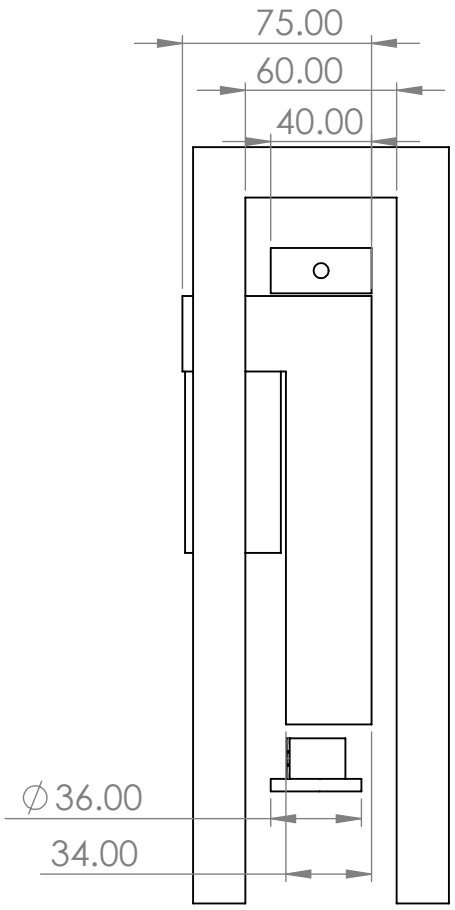
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: +-0.5mm ANGULAR: +-0.5 degrees	FINISH: 3D Printed	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION			

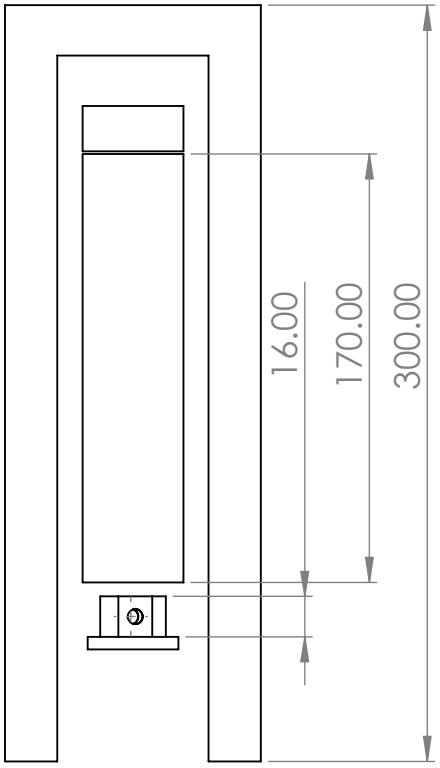
	NAME	SIGNATURE	DATE				TITLE:
DRAWN	CPC		6/5/2024				Powder Dispenser
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL: Red PLA			DWG NO. Dispenser-CPC
							A4
				WEIGHT:		SCALE:1:3	SHEET 1 OF 1

SOLIDWORKS Educational Product. For Instructional Use Only.









UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	3D Printed	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

PLA + Aluminium

DWG NO. CPC- Compressor

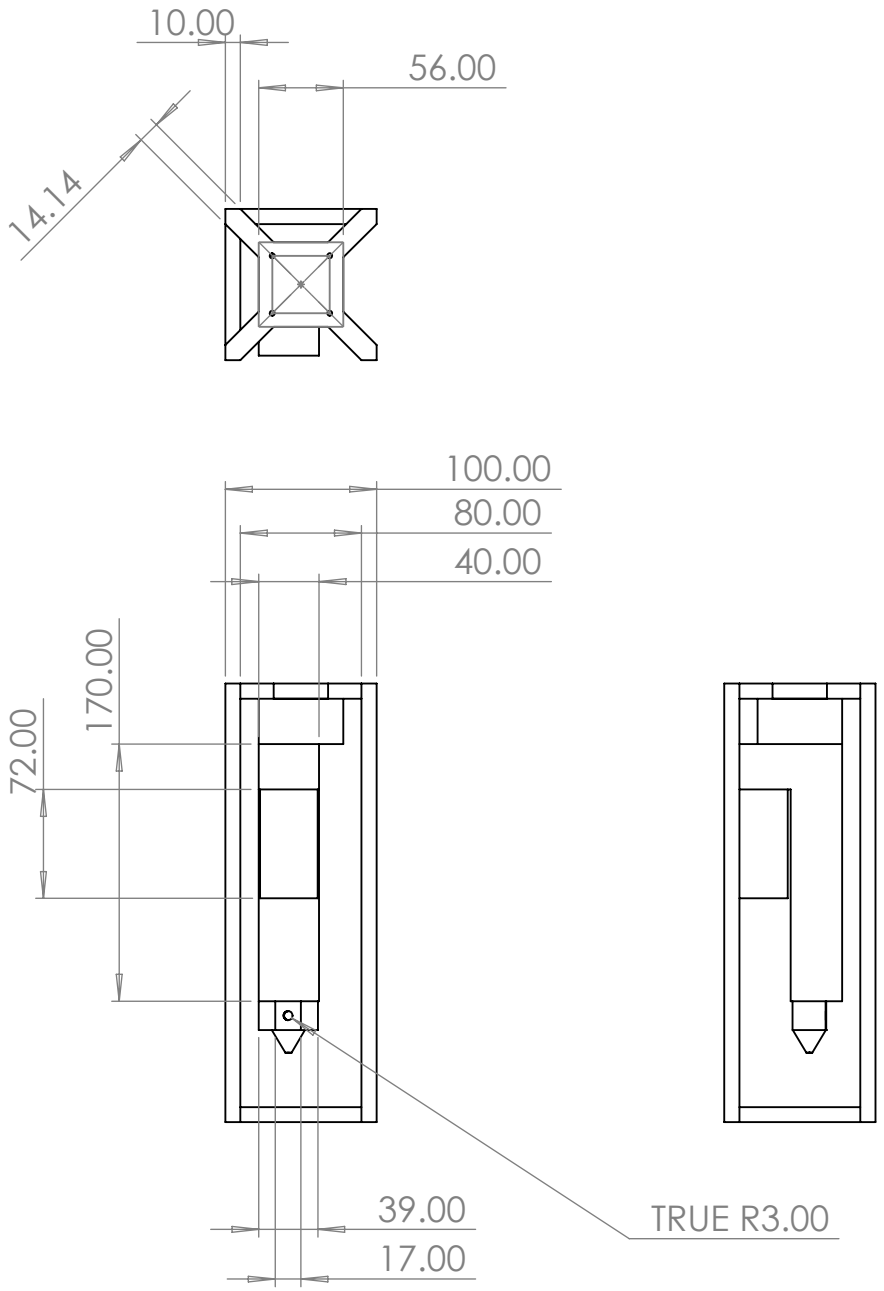
A4

WEIGHT:

SCALE:1:5

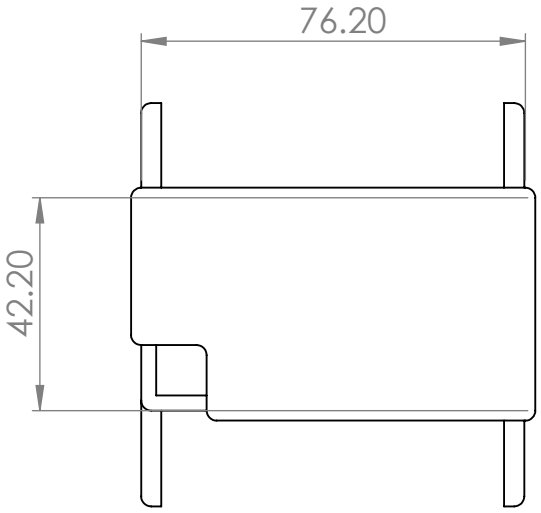
SHEET 1 OF 1

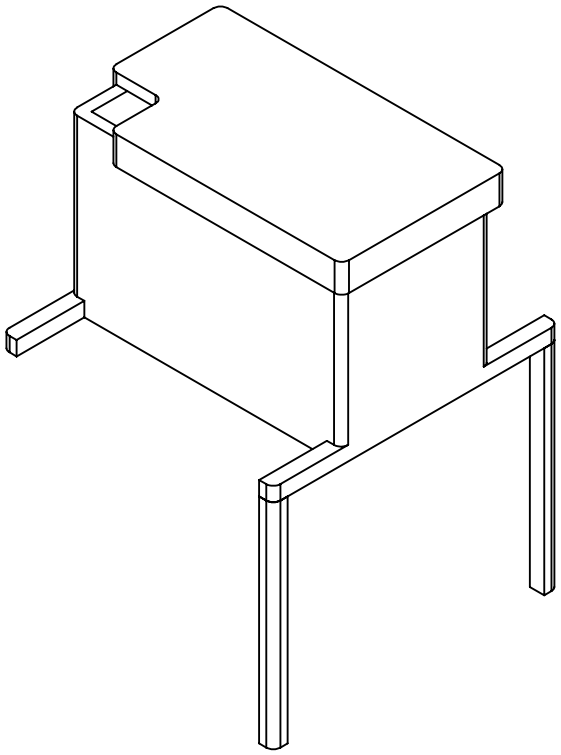
SOLIDWORKS Educational Product. For Instructional Use Only.

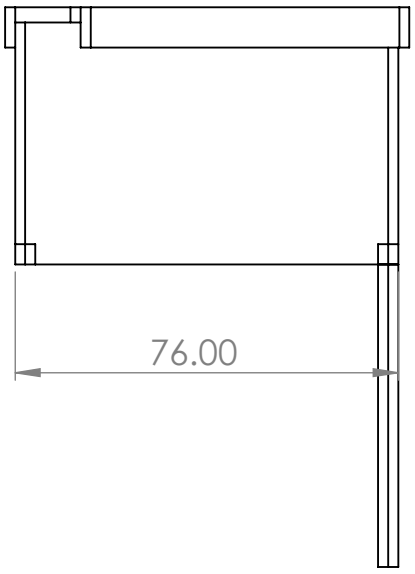


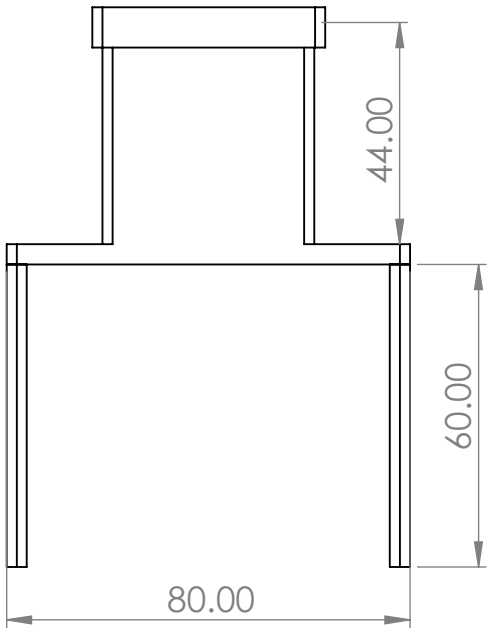
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ±0.5mm ANGULAR: ±0.5mm		FINISH: Factory		DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
	NAME	SIGNATURE	DATE			TITLE:	
DRAWN	CPC		4/05/2024			INDENTOR	
CHEK'D							
APPV'D							
MFG							
Q.A				MATERIAL: Metal		DWG NO; CPC-indentation	A4
				WEIGHT:		SCALE:1:5	SHEET 1 OF 1

SOLIDWORKS Educational Product. For Instructional Use Only.









UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	3D Printed	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

Camera Holder

PLA

DWG NO. A4

WEIGHT: SCALE:1:2 SHEET 1 OF 1

SOLIDWORKS Educational Product. For Instructional Use Only.