Embedded Systems Design Report

CN-SEEDER

Doncon, Aaron
18969254

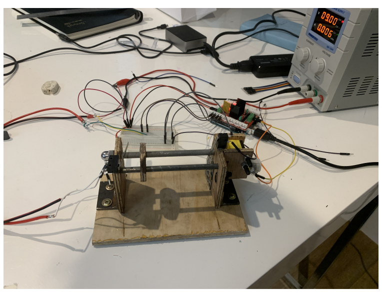
Morcombe, Alvan
19457572

Loos, Connor
19819473

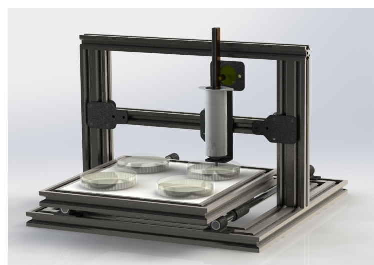
Huisman, Ian
14837056

Morris, Maxwell
19769552

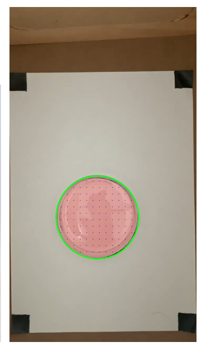
June 10, 2023



(a) Prototype/Testing Implementation



(b) Prototype Design Gantry



(c) Petri dish Seed Segmentation

Single Axis prototype using L298 H-bridge, Arduino and DC Motor demo here.

Contents

1 Executive Summary

Our product, an automated solution to the manual process of planting small seeds via an injection of aqueous solution into agar, seeks to optimise a tedious and repetitive step of botanical and crop research.

In our competitor analysis we determine this is a novel product with no direct competition.

We determine that a business could be built around this device focused by exploring different methods to generate revenue through subscriptions, apps, hardware and repair costs.

We note that this business is risky in terms of getting off the ground as it is a robotics company but the product would be to add value to customers if successful.

In this report we detail various prototype design implementations. One is a gantry designed in solidworks. Another component that is prototyped is the computer vision code for converting the video input into coordinates for moving the gantry it is able to successfully identify the petridish, determine a rough scale factor of pixels to cms to build a coordinate space and automatically determine seeding locations. We also implement a single axis of the gantry that can move the "head" to a specific location along the axis. Demo here: https://drive.google.com/drive/folders/1nk5fyYcYKY9E-X7IaWms3W8vrHqoJUqB?usp=share_link.

2 Problem Statement

The field of crop disease management is one where the solutions are found through rigorous physical testing in large batch volumes. This involves the physical placement of each seed, crop or specimen into sterile dishes, known as a Petri Dish. This is a laborious process as the main method of seed placement used by research facilities all over the world, is to 'place-by-hand'. If the seeds are too small, such as Arabidopsis Thaliana, a pipette will be used while the seeds are suspended in an agar solution. A current solution to using physical labor to fill the trays is to usea semi-automatic method of seed injection, this comes in the form of a Distriman. A semi-automatic adjustable pipette, which can eject a set amount at the push of a button. The reason why this is not the standard method of application is due to its steep price tag of 50,000 dollar, which is why the CN-SEEDER was designed. The CN-SEEDER is a fully automatic seed dispersal unit that was designed to suit the Arabidopsis Thaliana seed but is capable of seeding larger plants as well.

The main design is based around the concept of a 3-axis CNC Mill, where the mill is replaced with a seed ejector capable of detecting and releasing seeds from a reservoir into a Petri dish. The movement around the X-Z plane allows the CN-SEEDER to accurately place seeds in a uniform matrix pattern, with speed and precision that is not possible when using a pipette or Distriman.

The CN-SEEDER's movement in the Y-Axis is what allows the seeds to be precisely placed into an agar solution, at a customisable depth. The seed depth is uniform for all seeds and allows for an accurate and uniform plant growth.

3 Implementation Plan

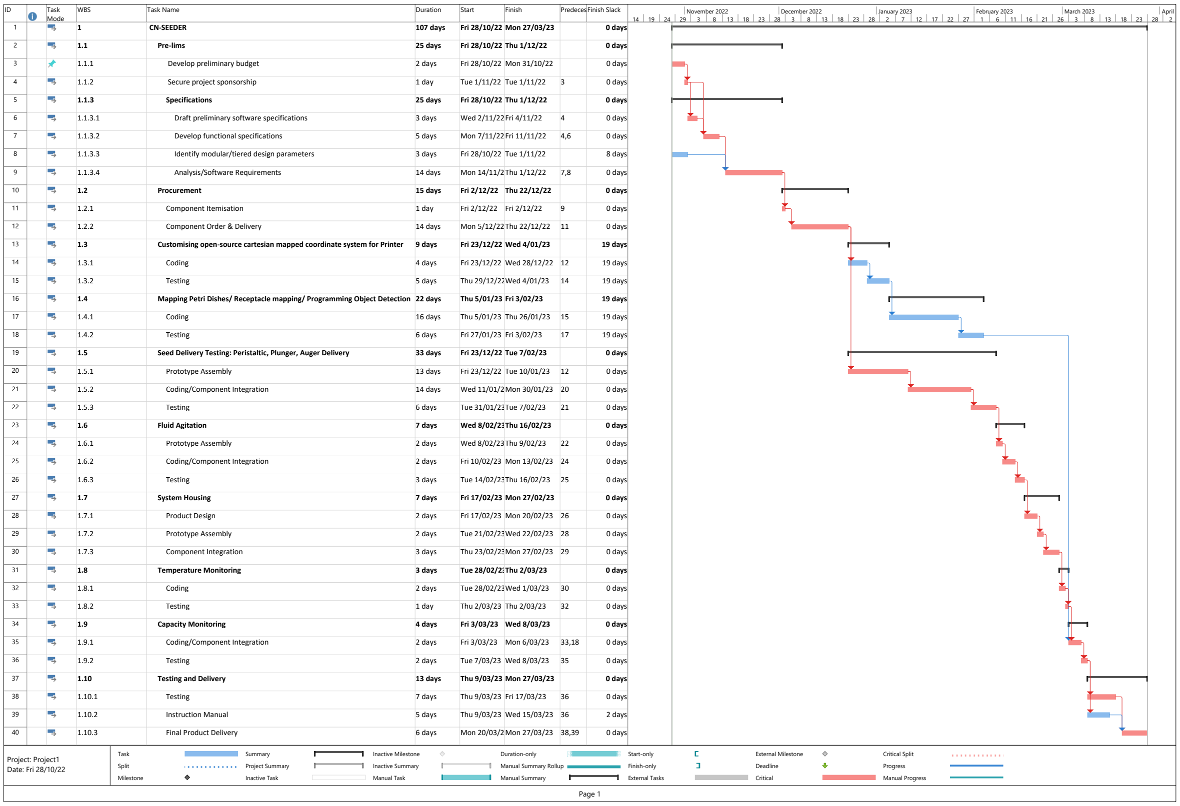


Figure 1: CN-Seeder Implementation Gantt Chart

3.1 Work Breakdown Structure

The work breakdown structure provides an overview of the projected tasks required to be completed to design and implement the initial working prototype of the CN-Seeder. These have been arranged in the milestones required to be reached to prove the feasibility and in order of preceding and subsequent tasks. Many of the tasks can be completed concurrently while others will require their proceeding steps before the subsequent components can be practically commenced. This breakdown structure will be used as a guide, and tasks may require longer or shorter durations depending on the complexity of the requirements.

3.2 Functional Components

3.2.1 Seed Delivery Method/Testing

The core component of our product is a functioning seed delivery method that is both functional, accurate, reliable, easily serviceable and efficient. As such the longest duration was assigned to this component, and the greatest level of complexity is foreseen in the completion of this task. Once the initial components are procured and delivered to the team, one of the initial tasks that has to be undertaken is in finding a feasible method of delivery that is able to detect a seed suspended within the agar solution and is able to deliver this through the printer head to the agar petri dish.

3.2.2 Fluid Agitation

The means of delivery that has been selected for our product requires that the Aradopsis seeds remain suspended in a suitably viscose agar solution at a concentration that ensures that minimal solution is required for each injection into the Petri dish. As such to ensure that the seeds remain suitably suspended throughout the solution constant agitation will be required, which will necessitate the design and implementation of a suitable housing unit and means of agitation that will maintain the solution at the desired levels of suspension to ensure that the seed delivery system is receiving seeds in the solution at the required rate and with minimal excess solution. A viable means of agitation will ensure that the product is able to deliver seeds in a variety of solution concentrations and viscosity and considerable time may be required to find a practical working model.

3.2.3 System Housing

To ensure the deliverable of a viable product to market, the CN-Seeder must be compact, functional, easily serviceable and durable. As such, the system housing design will be commenced after the major components have been designed and tested to ensure that the final model is created around these components to also produce a visually appealing product. This will also ensure that the space required is minimised to reduce unnecessary design features to reduce the overall cost in both its assembly and to the consumer.

3.2.4 Temperature Monitoring

The inclusion of temperature monitoring for both the internal circuitry as well as for the surrounding environment will ensure that the final product is operating within expected parameters and heat will not have any undue effects on either the viscosity of the agar solution, the petri dishes or the seeds themselves.

3.2.5 Capacity Monitoring

The inclusion of hardware to monitor the remaining fluid within the printer will ensure that there is enough solution remaining, with the correct seed dispersion to be able to accurately complete the required number of prints. An alert system will also be required to notify the user if there is not enough solution or seeds to complete the session, and this sensor will have to be placed in such a way that the sensor input will not become blocked or effected in a way that will give false positives.

3.3 Software Requirements

A combination of 'python' and 'c' is used to program the CN-SEEDER functions, as the axis movement is controlled by a 'c' using a set of 'c++' libraries. The image recognition scripts are ran on 'python' with the use of the OpenCV library.

3.3.1 Cartesian coordinate Customisation

Open source software will initially be used as the foundation for the printing system designed. This software design can be commenced before the initial delivery of the hardware components and will be customised to suit the requirements of our product. From the initial design of our product a Cartesian coordinate system has been identified as the best suit and will be modified as required throughout the design process until the final working prototype has been completed.

3.3.2 Object Detection

A camera attached to the print head will be used to identify the print region and to map out the seed dispersion on the receptacle placed on the print tray. Due to the potential complexity of this task, considerable time has been allocated to this task, and the use of machine learning will be incorporated to maximise the efficiency of the process in identifying the area and depth beneath the print head.

3.3.3 Seed Delivery Coding

Once the hardware component of the seed delivery system has been established and the interface and components identified, the hardware components can then be coded into the microcontroller and tested to ensure that under standard operating conditions the system will accurately and reliably deliver seeds through the print head into the agar. The coding and testing components together will account for a considerable duration in the design phase and may be a source of lag for subsequent phases.

3.3.4 Fluid Agitation Coding

Once a final model has been developed for the fluid agitation component, the requisite variables and conditions will need to be coded into our product to ensure that the solution meets the requirements of our product, and that the solution maintains the conditions that are established. This will need to be done in a way that maximises the efficiency of the system and that excess energy is not consumed in maintaining the solution. As a variety of solution concentrations may be required this is an area that may cause delays in the final product release as it is an area of increased importance to the final product.

3.3.5 Temperature Monitor Coding

The final product will utilise both direct and indirect temperature sensors, and these will be designed and installed in areas of high energy use and as passive sensors to ensure that the printer is operating within the desired ranges of temperature that will not affect the accuracy or integrity of the unit. Failsafe's will then be designed into the final product to either alert the user of unsafe or unfavourable conditions, cease operation or to recommence a print.

3.3.6 Capacity Monitor Coding

The sensors used to ensure that there is adequate seed solution for the required print will be coded into the microcontroller to provide alerts should there not be enough solution remaining or to cease a print job once all fluid has been expended.

3.4 Additional Requirements

3.4.1 Testing

To ensure that after repeated and ongoing use the CN-Seeder will continue to accurately and reliable operate a significant duration of time has been allocated to test, troubleshoot and resolve any unforeseen issues with the product once all components have been integrated into the final product. This is a key stage in the product development as the interaction of all systems may create conditions which do not arise independently.

3.4.2 Instruction Manual

Providing a comprehensive instruction manual on the setup, use and operating conditions required for our product will ensure the long term success of the CN-Seeder. A comprehensive guide also reduces the need for support to the consumer and should increase consumer confidence in our product. This has been left to the final stage of our product design to ensure that the content of the instruction manual accurately reflects the components used, and can include trouble shooting guides from any issues identified in the testing phase.

3.5 Final Product Delivery

Once the final prototype is completed a targeted product release and advertising campaign will ensure that our product will meet our companies fiscal expectations. The product will only be launched once a viable working prototype has been completed and associated collateral and instruction manuals finalised. This will be done in conjunction with industry partners, and working with both domestic and international markets to ensure the broadest reach of interest in the final product for the smallest expenditure, while also not exceeding our manufacturing capacity

4 Functional Design

Our product is designed to enable automatic and fully autonomous management of micro plant experiments such as *Arabidopsis thaliana*. The intention of the product is to minimise lab labour costs by automatically doing the following experimental steps:

- Creating seeding fluid
- Putting Agar on Petri dishes
- Inventory Management of Petri and Agar Dishes
- Seeding of the Agar Plants
- Sensor Management
- Experimental Data Collection

Whilst in order to turn this into a business we would have to implement each of these individual components, for this design report we have choosen to focus on the automatic seeding of the agar plates.

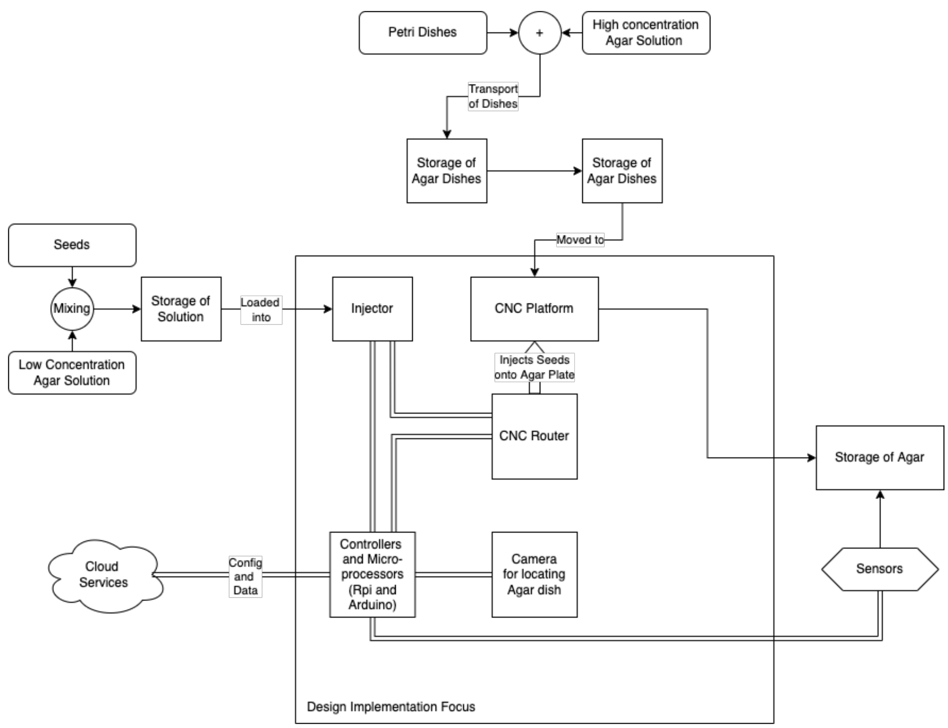


Figure 2: Overall Functional Design of System

5 Software Implementation Specification

The software for this project will enable the system to automatically locate the petri dishes and control the CNC router so that the seeds are correctly placed on the plate.

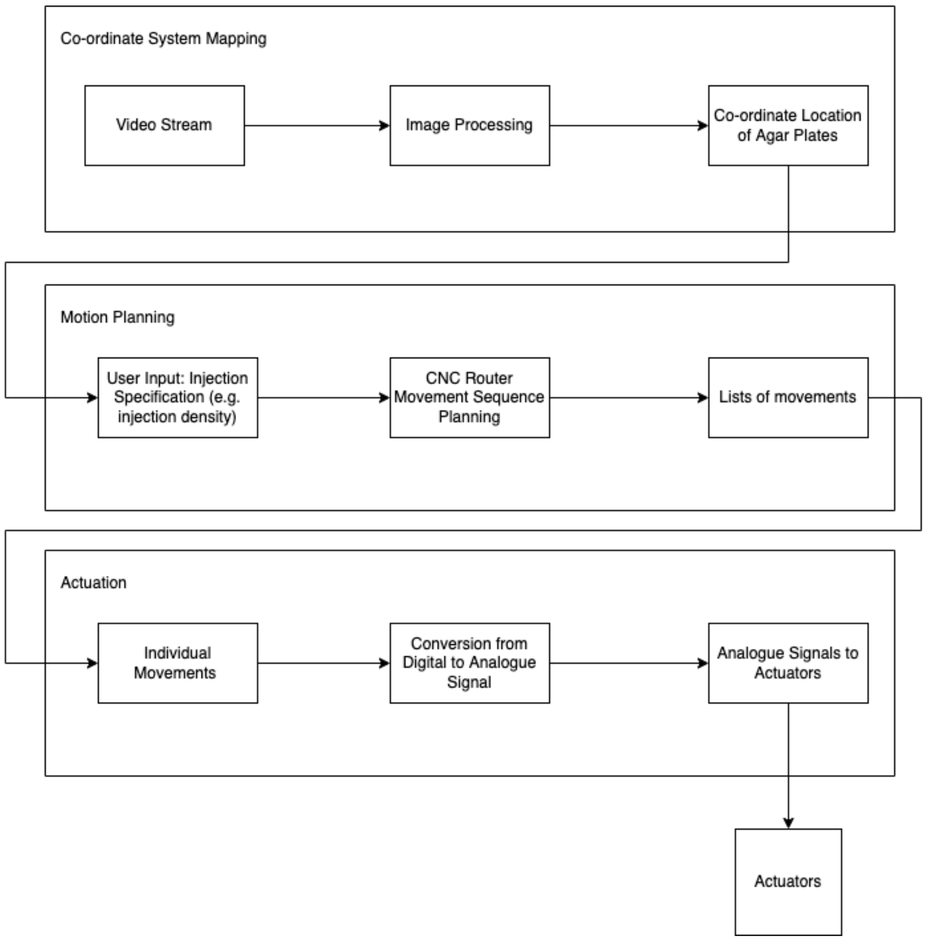


Figure 3: Software Specification Diagram

6 Hardware Components

Here we review each hardware component going into our design.

6.1 Control and Sensing

Device Name: Raspberry Pi 4
Overview:
A raspberry pi is a mini computer that has a microprocessor. It has significantly more computing power than a Arduino. We will be using the raspberry pi for image processing and communicating the location of objects to the arduino. **Hardware Specific Information (from datasheet):**

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 5V DC via GPIO header (minimum 3A*)
- 2-lane MIPI CSI camera port
- Raspberry Pi standard 40 pin GPIO header

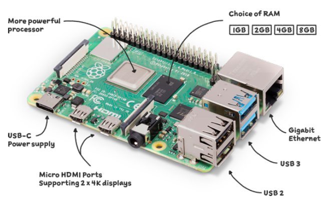


Figure 4: Raspberry Pi 4 model B. Adapted from [1]

Device Name: Arduino Uno
Overview:
The Arduino Uno is a cheap microcontroller. The advantage of using the Arduino over the R.Pi for interfacing with the the H-bridge is that it can drive a higher voltage (5V vs 3.3V) on the H-bridge. This could potentially be useful as it lets us increase the saturation voltage on the input to the H-bridge's transistors.
Hardware Specific Information (from datasheet):

- 14 DIGITAL I/O (of which 6 provide PWM output)
- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Clockspeed: 16MHz
- Up to 9mA sink. With $100k\omega$ pull-up resistor optional.

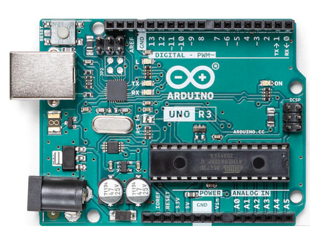


Figure 5: Arduino Uno R3. Adapted from [2]

Device Name: Raspberry Pi 5MP Camera
Overview: This camera is used as a sensor to detect where the petri dish is on the the CN-SEEDER base and to calibrate the control systems coordinate system. The camera outputs at up to 1080p at 30fps which is near the limit of what a raspberry pi can process in opencv. **Hardware Specific Information (from datasheet):**

- 1080p @ 30fps, 720p @ 60fps and 640x480p @ 60/90fps.
- S15-pin MIPI Camera Serial Interface.
- 25(W) x 20(L) x 9(H)mm.

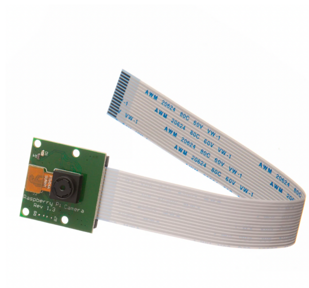


Figure 6: 5MP Raspberry Pi Camera. Adapted from [3]

Device Name: Z6343 L298 H-Bridge Motor Shield For Arduino
Overview:
The L298 h-bridge chip is designed to drive inductive loads like DC Motors. It can be use to control the voltage over two motors and can handle up to 2A per motor at 46V. This is signifcantly more than what is provided by our arduino. It effectively acts as a control buffer for the arduino to the motor. **Hardware Specific Information (from datasheet):**

- Overtemp protection.
- Operating Supply voltage of 46V
- High Noise input immunity
- Total DC of up to 4A
- Low saturation voltage for increased input range from arduino (-0.3 - 7V)

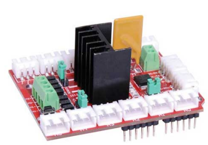


Figure 7: Z6343 arduino shield with L298 chip[4]

Device Name: SPDT 250V 3A Miniature Micro Switch
Overview:
Limit switchs used on either side of the Axis. The NC functionality was not used as the switches were in pin-gnd configuration. **Hardware Specific Information (from datasheet):**

- Shorten-able length
- Solder Connection
- Low resistance triggering.

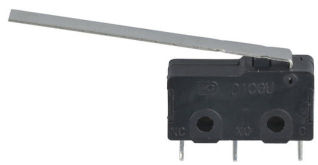


Figure 8: SPDT Microswitch image adapted from [5]

Device Name: J0016
Overview:
48:1 gear box to increase torque on rotating the shaft. **Hardware Specific Information (from datasheet):**

- 1.5A stall
- 3-6A operation. (Works fine at 17V)

Figure 9: J0016 image adapted from [6]

Device Name: RS PRO Photoelectric Sensor, Receiver
Overview:
Sensor uses a sender and receiver to detect laser obstruction.

- 10-30v required to operate
- <25 mA of current consumption
- <5% repeated accuracy
- 8.2ms response time

Figure 10: RS PRO Photoelectric Sensor, Receiver image adapted from [7]

Component Name: T-Slot Photo Interrupter
Overview:
A simple plastic sensor with two elements - an IR LED and an IR photo-transistor, situated across a U-shaped gap.

- 5-12V Logic power in (VCC - brown lead)
- Configurable LED input (L - pink lead)
- Transistor output from open collector, requires pullup resistor (OUT - black)

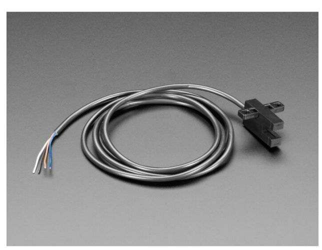


Figure 11: T-Slot Optic Sensor, image adapted from [8]

6.2 Frame and Actuation Components

Component Name: Square T-Slot Frame Rail
Overview:
20x20 sections easily fastenable via angled slots on all four sides. Material is die-extruded 6063 T-5 Aluminium.

- Ultimate Tensile Strength of 186MPa
- Fatigue Strength of 68.9MPa
- Standardized dimensions and wide acceptance in 3D printing and other projects requiring quick modular construction.

Figure 12: Extruded aluminium frame sections image adapted from [9]

Device Name: Stepper Motor
Overview:
The stepper motor will be used to control seed output via rotating gears.

- 12V rated voltage
- 0.33A maximum current draw
- 2.3kg*cm full load torque
- 1.8 degree step angle

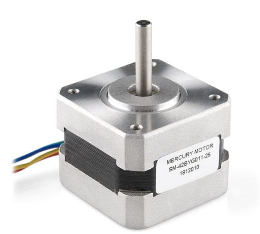


Figure 13: 12V Stepper Motor, image adapted from [10]

7 Hardware Implementation

Here we show the assembly of our components into a finished product.

7.1 Motors

DC Control:
The H-Bridge is connect using the below schematic. It should be noted that in software `MIn` and `MIp` pins are always inverted. The H-Bridge can easily be seen here to be acting as an amplifying switch. It should also be noted that if `MIn` and `MIp` were not inverted that there would be short circuiting between Q2 and Q4 or potentially Q1 and Q3.

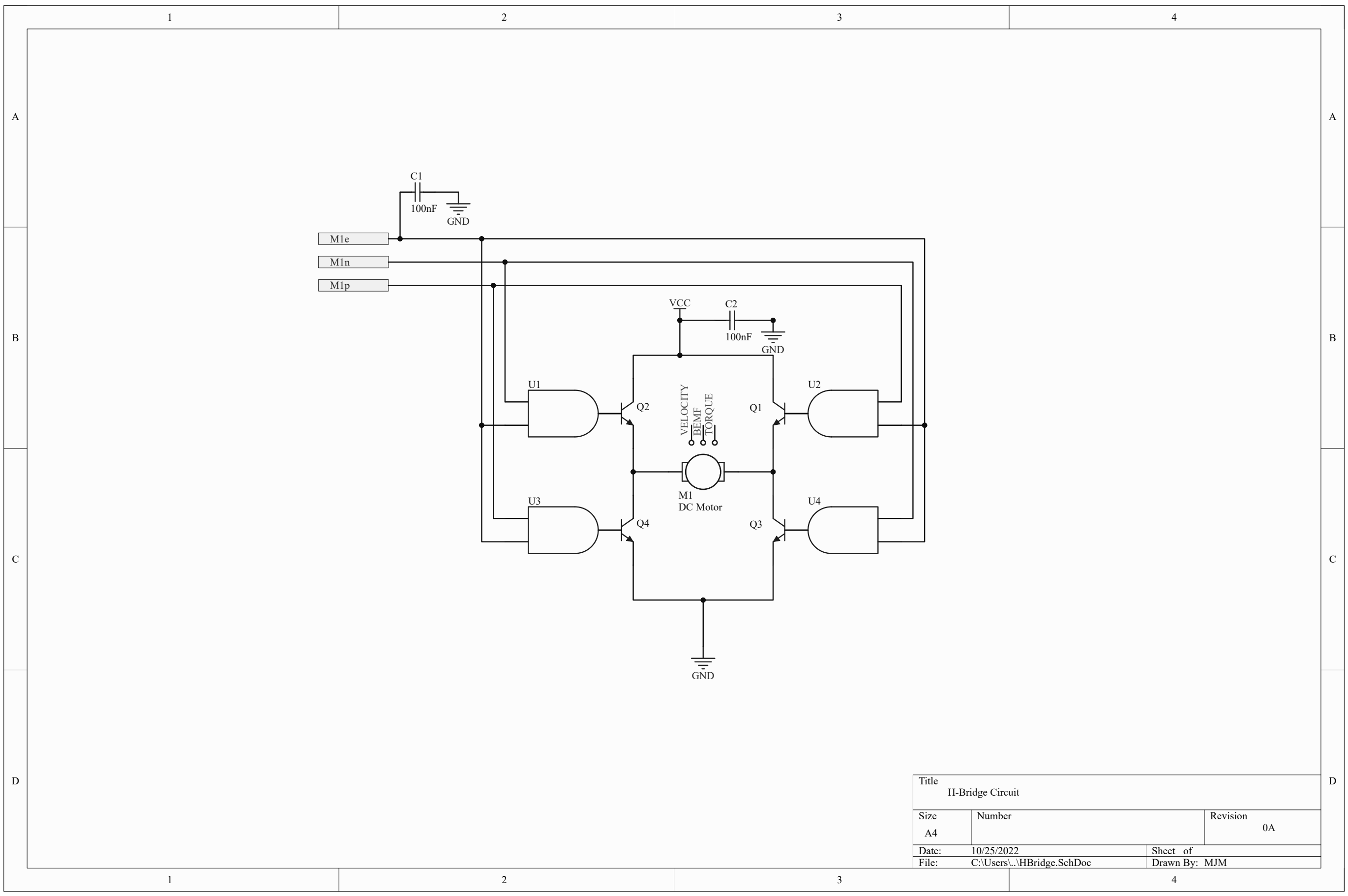


Figure 14: H-Bridge schematic indicating the input pins with their software references for the axis prototype.

Stepper Control:

7.2 Frame and Actuation

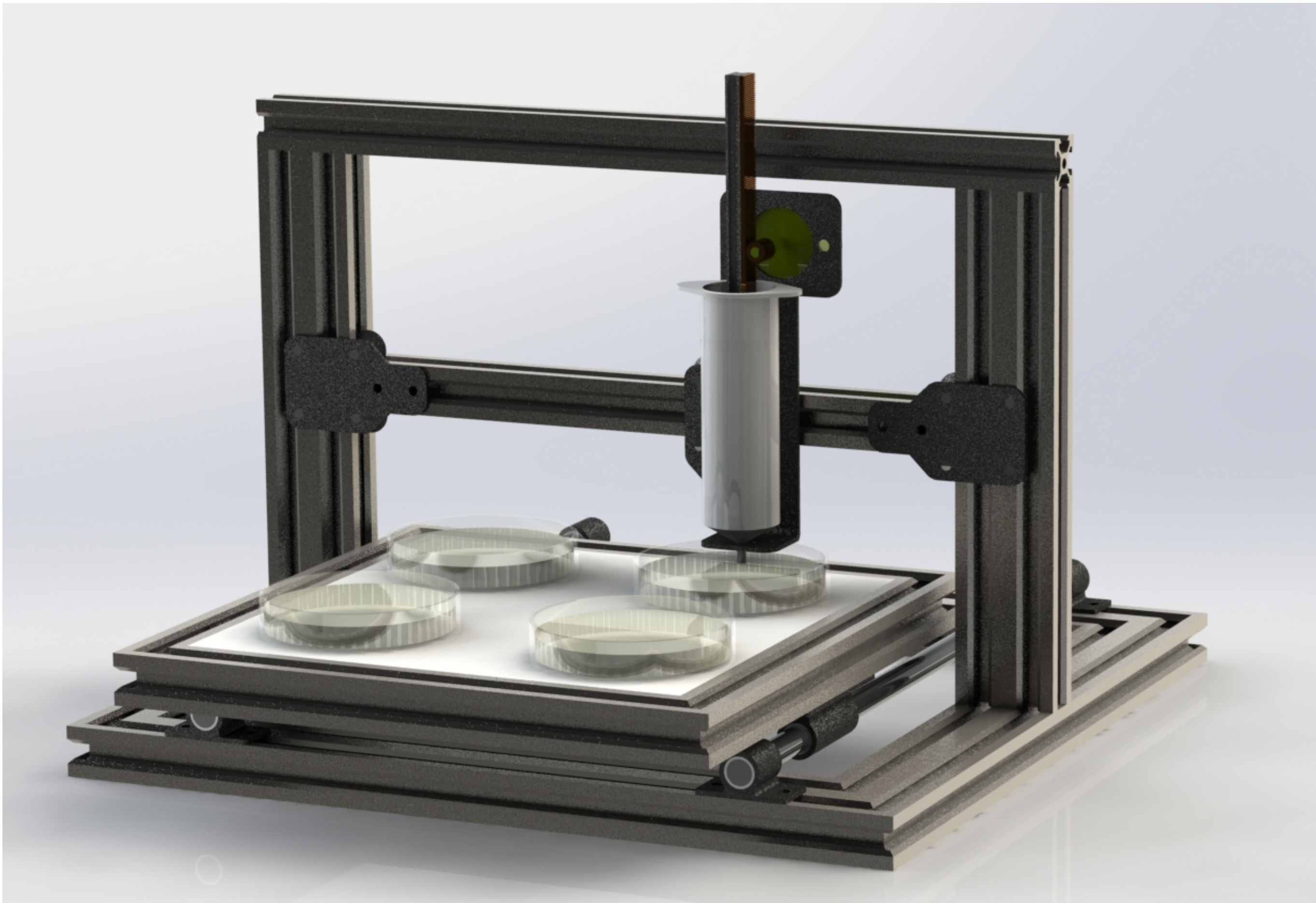


Figure 15: Final Render of the assembled Gantry and Seed Head.

Material:
Using the modular frame rail design ([9]) and accompanying fasteners, our team are able to quickly and easily achieve translation/actuation of the separate frame pieces. The T-slot design features sloped channels along each face of the square section, meaning a V-shaped roller wheel will auto-centre itself while travelling in a fixed dimension. This design lends itself well to a Cartesian system of movements, where any movement in 3D space is expressed as a change in position along X, Y, and Z co-ordinates. **Coordinate/Movement Layout:**
The design choice was made to assign a single dimension of movement to the seedbed (platform affixed with petri dish/es), while the seedhead (dispenser) would move in the remaining two dimensions. This approach has a number of benefits for our specific design:

- Reduces number of frame components required. Giving the head 3 degrees of freedom means the upright components of the frame must occupy those three dimensions fully, leading to a cube layout of the frame. Removing one dimension of movement from the head means the upright section takes the shape of a 2D square arch, saving on components.

- The more degrees of freedom each actuating part has, the higher the likelihood of the "play" (unplanned movement caused by loose-fitting components) from different elements of actuation combining and having unexpected effects on operation. The translation of the seedbed is kept separate from the translation of the head, both being mounted to a fixed heavy-duty base.

- Aiding the Stretch Goal of automatic loading/unloading of the device. A horizontal-translating base can easily be swapped out for the conveyor design laid out in the section **Stretch Goals - Automatic petri dish loading and unloading.**

Drive:
Once this system of movement had been established, the next major articulation implementation decision was the driver of movement (e.g. how to link the motors to the sections of frame to be moved). A system of belts, affixed to the driving motors with toothed belt gears, was chosen. Although a popular choice for printers (both 2D and 3D) as well as much other single-dimension actuation in both small and large machinery, it had to be compared and contrasted with Ball-Screw linear actuation.
A Ball-Screw linear actuator uses a nut with recirculating ball bearings that rotate around a square-ground screw thread. The rotation pushes the bearings around the thread, either raising or lowering them with the nut in tow. Although this design is used in many 3D printers, CNC machines, and other devices requiring high-precision movement along an axis, the belts were chosen for a number of reasons:

- Cost. Belt drives are cheaper due to plastic rather than metal construction, less total material, no grinding/machining required, and lower tolerances in regards to the fit and finish.
- Ease of implementation. The ball screw system requires the motor mounted to the end of each threaded rod, and the whole package needs to sit directly in-line with the intended axis of movement. In contrast, the belts can be looped around runners, tensioners, or geared drives. The motors can be places anywhere on the device the belts can easily be routed to, and the size of the device and the light load on the belts mean there are less drawbacks (friction, wear, excess lateral forces etc) to mounting the motors further away from the components they are to move.

Control/Edge Sensing:
Although motor control itself is managed between sections **Hardware Implementation - Motors** and **Software Implementation - Embedded Control Code**, the device needs at least basic input for start and end points of each axis. Although this could be hardcoded in and a software solution used to keep track of current position, this would make the device prone to failures. Any loss of power/wipe of onboard memory would result in the device having no reference of position and potentially attempting to push components past their furthest position with catastrophic results. It would also make any maintenance exceedingly difficult, as any moving component in the device would have to be detached and reattached in an identical position.
The simplest and most widely-used solution is to use an optical edge sensor. Small and cheap slot sensors feature a U-shaped plastic frame with a photodiode on one end and a small LED in the other. The name slot sensor comes from the fact the LED points out a narrow slot facing the photodiode, meaning it will return a low value only when the small path of light is physically interrupted. Unlike other optical sensors, this simple and cost-effective design has a low chance of false positives (outside interruptions) and false negatives (outside light hitting the photodiode) due to the small gap and controlled beam of light. A sensoe sitting at each end of the three axes of movement (6 total) would give the controlling software a range of values for each axis to move between.

7.3 Feeder-Seeder (Seed ejection component)

The seeding process will be done through a syringe which stores an agar/seed solution and has a black spot sensor at its ejection nozzle to detect weather a seed has been pushed out. The plunger of the syringe will be controlled by 2 servo motors with teeth on the on each side of the plunger. The plunger will also be toothed as the motor teeth will grip onto the plunger to move it. The syringe will be fitted into an L bracket which will have a hole in the lower part of the L to fit the syringe nozzle into, as seen in the following figure. The plunger will then be inserted into the top preventing movement of the syringe. 3D printer will move to its first position and the motors will turn, pushing the plunger down till the sensors pick up a seed passing. One of the sensors will output laser and the other sensors receives and reads light. Once the seed passes through the syringe nozzle, the initial laser will be blocked and the light sensor will pick up on this, pinging the embedded system that a seed is ready to be expelled. The motor will keep rotating for a set amount of time to push out the seed the sensor spotted. As the sensors are placed slightly before the syringe expulsion point (as seen in figure x). During this point the sensors will still note of any seeds passing and their position relative to the expulsion point. Below the seeder feeder can be seen.

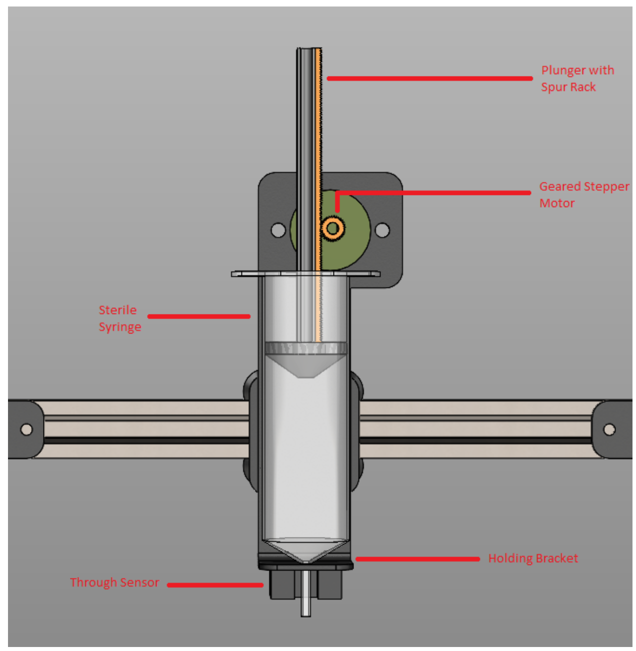


Figure 16: Seeder Feeder

Given the expected mass of the seedhead and related electrical components, plastic, although easy and cheap to work with, would not have the required strength to meet the rigidity requirements.

8 Software Implementation

8.1 Co-ordinate Mapping (Localization)

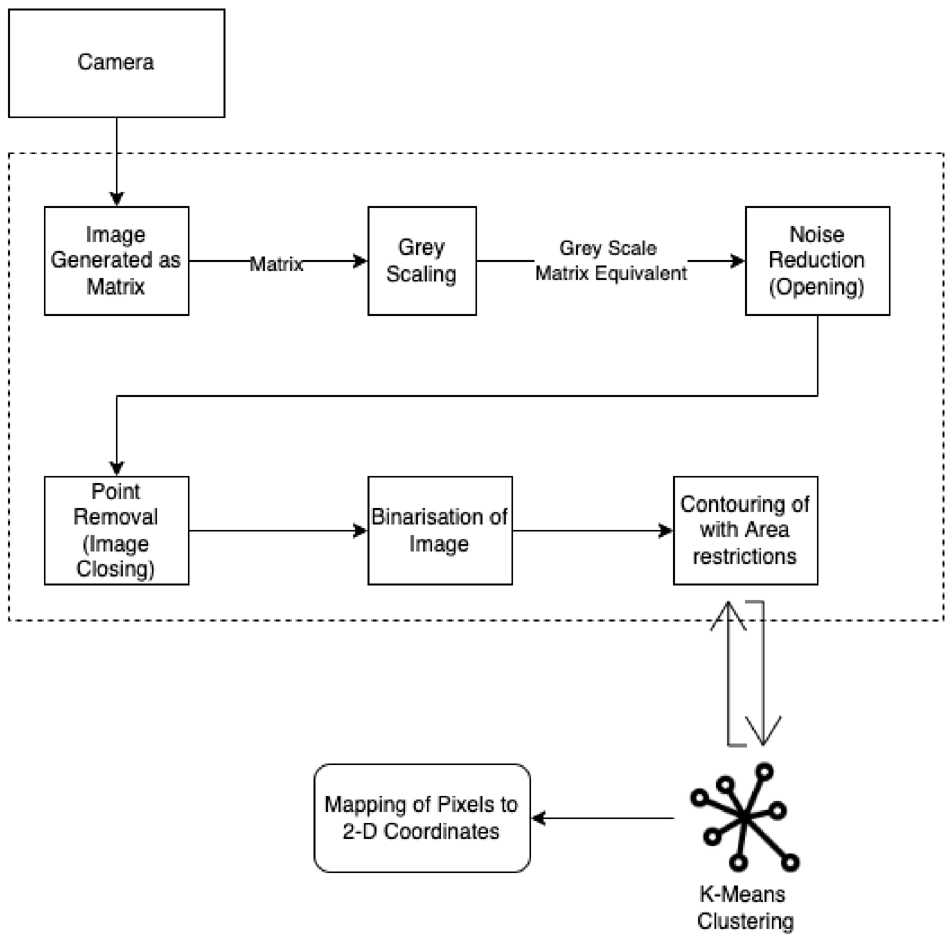


Figure 17: Reference point mapping

In order to send the distriman attachment to insert the seeds into the petri dish we must first build up a coordinate space. The coordinate space is required so the robot can localize itself with reference to the petri dish. The base of the CN-SEEDER will have distinct marking on it that can be detected by the camera. In this proof of concept we use black squares on a white background. The distance between the points are know (26cm and 17cm) and this is will be used in the next step to generate location coordinates of the Petri Dish.

1. Camera Input is converted into a opencv `Mat` a multi dimensional array representing the image
2. The input image is converted into another colour space (grey scaled) using a colour space transformation
3. Noise reduction is done by using an "opening" by applying a centred 8x8 rectangular structured element. (Simply just an 8x8 matrix of ones that will be ANDed with the image). This removes any bright spots in the image.
4. Then holes inside of the shapes are filled by applying a closing structured element.
5. Then the image is "binarised" based on a certain threshold colour values are turned into either black or white

6. Then a contouring function is used to try and find 4 sided shapes the above process is repeated over 10 seconds to build up a collection of where the reference markings are, this is then put into a 4 point K-Means clustering algorithm, which returns the 4 centroids which will be approximately the centres of the markings.
7. These co-ordinates are used to create the mapping. As the distance between these points is known and the pixel coordinates are also know a coordinate space can be given.

There could potentially be a better method of accomplishing this but we found that using a brute force method to have a reasonable tolerance to low light conditions. See figure 19

8.2 Locating the Petri Dish and Determining Seed Location

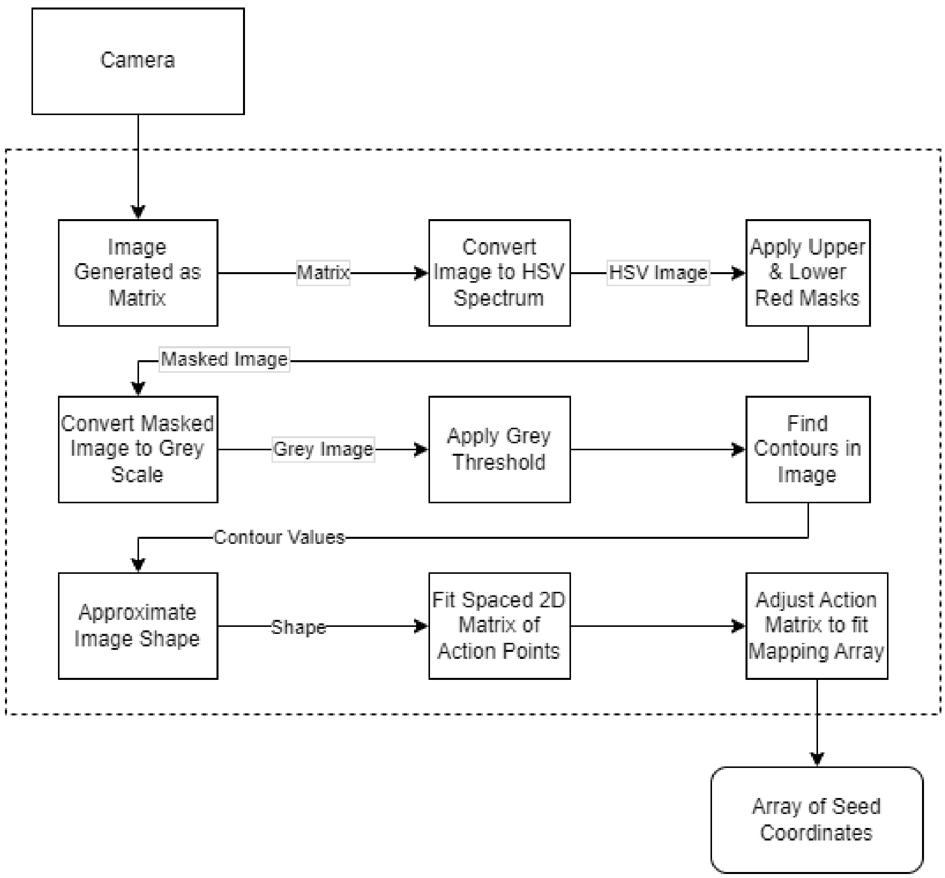


Figure 18: Finding Petri Dish and Determining Seed Location

In order for the distriman attachment to know where seeds should be placed in the petri dish, the CN-SEEDER must know the location of each seed, in its own coordinate space. In practice, the top of the Petri dish is painted red. This gives a high contrasting colour, which can easily be detected against the agar or any other background materials.

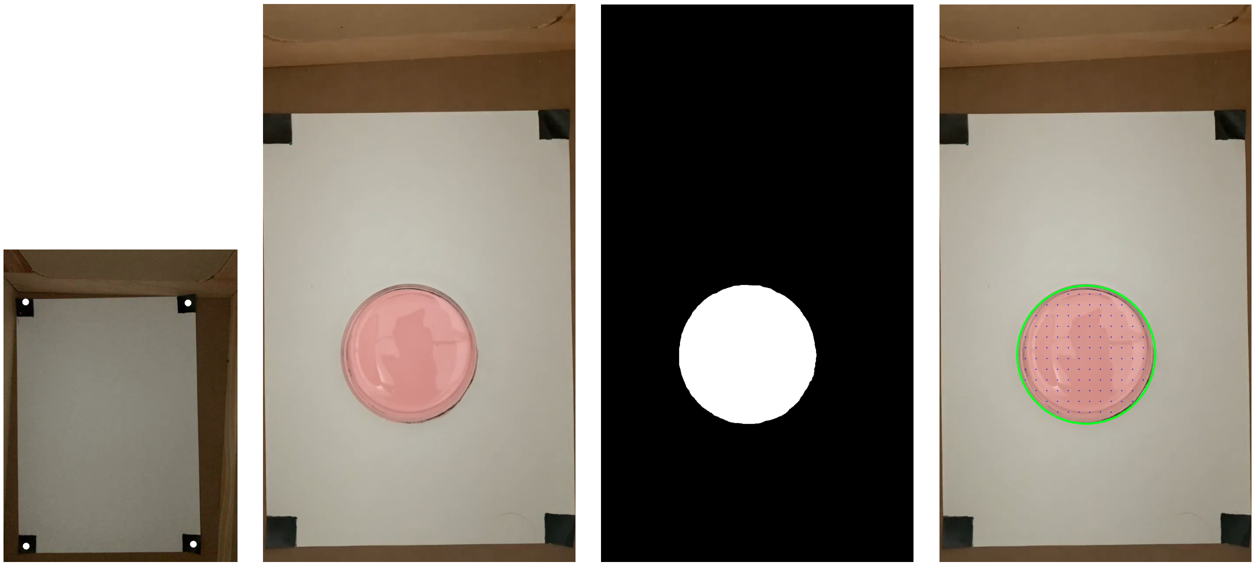


Figure 19: Coordinate mapping on first image from left.Right of that is the Petri Dish and Seed Position Image Recognition Algorithm.

1. Camera Input is converted into a `Mat` a multi dimensional array representing the image.
2. The image is converted into the HSV color space.
3. A red mask is applied over the image. This removes any colours that are not red, leaving only the shape of the petri dish.
4. The image is converted into the grey colour space.
5. A grey threshold is then applied to the image, ensuring a crisp and clear representation of the shape of the petri dish is left.
6. Contours in the image are then found using the OpenCV function “findContours”.
7. The shape of the Petri dish is than determined by using the OpenCV function “approxPolyDP”
8. Depending on the shape of the dish, circular or rectangular, different methods in generating the array of points will be taken. Both methods output a 2D array of coordinates “Action Points”, in reference to the image using pixels.
9. The array of Action points is turned into real usable coordinates by taking the mapping coordinates of the CN-SEEDER, determined by the “Coordinate Mapping” software stage. Since the same camera has been used to both calibrate the CN-SEEDER and photograph the Petri dish. We can take the coordinate system of the CN-SEEDER and fit it over the array of pixels derived in step 8, allowing the location of seeds to be determined in the same coordinate system as the machine itself.
10. The seed array is then passed to the controller of the system.

```python
img1 = cv2.imread(image)
display_img = img1.copy()
img1 = cv2.GaussianBlur(img1,(9,9),0)

img_hsv=cv2.cvtColor(img1, cv2.COLOR_BGR2HSV)

lower_red = np.array([0,60,60])
upper_red = np.array([15,255,255])
mask = cv2.inRange(img_hsv, lower_red, upper_red)
cv2_imshow(mask)

  circles = cv2.HoughCircles(mask,cv2.HOUGH_GRADIENT,1,20,param1=50,
                             param2=15,minRadius=150,maxRadius=170)
  # Draw circles
  if circles is not None:
      circles = np.round(circles[0, :]).astype("int")
      for (x,y,r) in circles:
          cv2.circle(display_img, (x,y), r, (36,255,12), 3)
  if circles is None:
    print("No Circles found")

space_mm = 7 #mm
xC=x #x-centre
yC=y #y-Centre
space=math.floor(((r*2)/90)*space_mm) #round down
xPoint=0
yPoint=0

dims = img1.shape
xPoints = []
yPoints = []
for i in range(dims[1]):
  if(xPoint<=dims[1]):
    xPoints.append(xPoint)
    xPoint=xPoint+space

for i in range(dims[0]):
  if(yPoint<=dims[0]):
    yPoints.append(yPoint)
    yPoint=yPoint+space

for y in yPoints:
  for x in xPoints:
    distToCenter=math.sqrt((math.pow(x-xC,2))+(math.pow(y-yC,2)))
    if(distToCenter<(r-5)):
      display_img = cv2.circle(display_img, (x,y), radius=0, color=(255, 0, 0), thickness=2)
```

8.3 Seed Feeder Mechanism

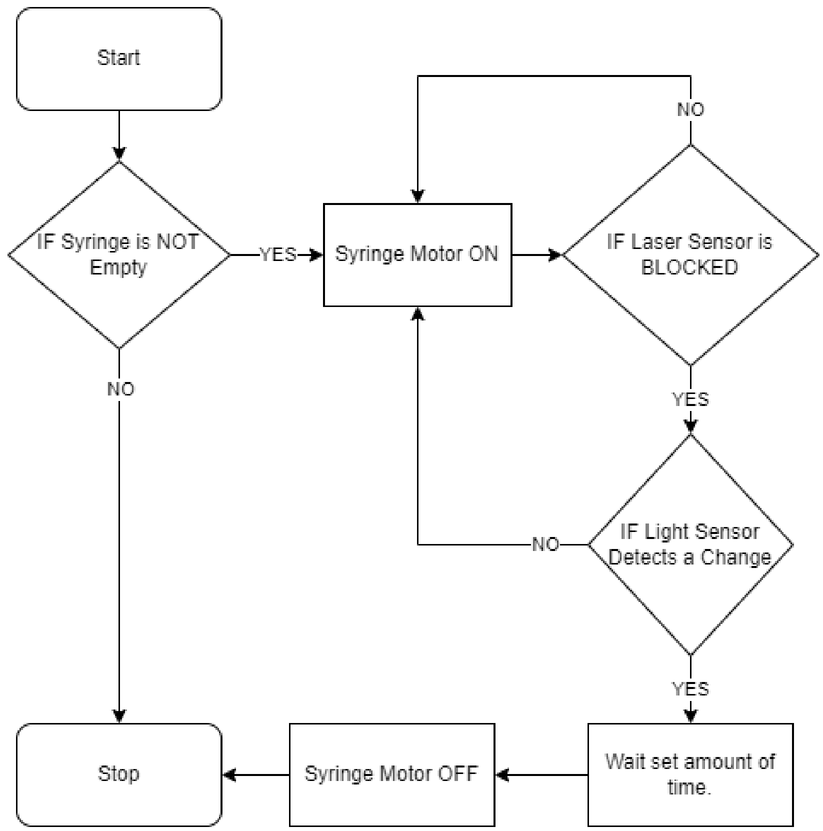


Figure 20: Seed Feeder Flowchart

In order for the seed feeder attachment to function correctly, we must assume the feeder is already in the correct position for dispension. Unlike the previous systems, the feeder is almost all mechanical as it requires no external libraries to run, only simple if-statements for sensor comparisons and state checks.

1. Upon receiving a signal to start the process, the feeder will check if the syringe is NOT empty.
2. The syringe motor is turned on until the laser sensor is blocked and the light sensor detects a colour change.
3. The system waits a set amount of time.
4. The Syringe motor is turned off.

8.4 Embedded Control Code

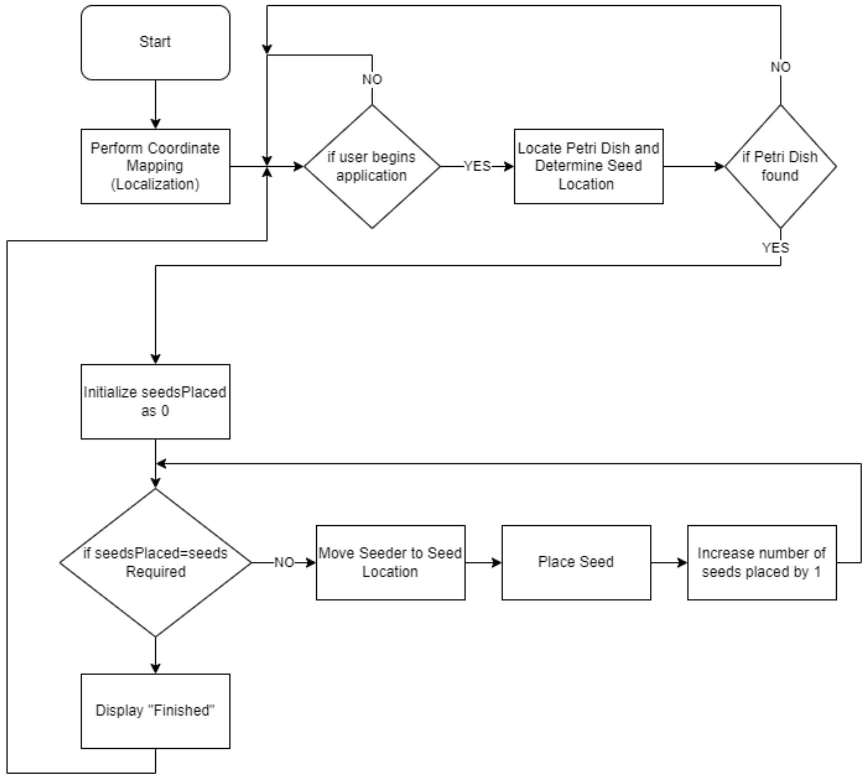


Figure 21: Controller Flowchart

This control code is for:

- Setting up the Axes. Calibration.
- Controlling the Axes (tell them where they need to be in space)

Calibration is getting inputs from the real world so that the each axis knows it's location in space. By know the time it takes to get between each end of the axis the speed can be determined. Whilst there are some issues with this method such as needing to account for acceleration and calibration drift, it is a cost effective method of keeping track of position. Some applications such as printers use optical tape and wall paints use optical spatial tracking. We opt out of using these methods as if your axis is made out of high quality ACME thread you can get good enough tracking for our purposes just by getting a simple initial speed calculation. One downside is that every so often the axis will need to know how to recalibrate.

In our implementation we have two prototypes using both limit switches and photoreceptors. Both behave similarly. The motor drives the axis point to collide with on side of the axis. When it collide it stops. Then starts timing the time to get to the other side.

To implement our design we use a H-bridge. A h-bridge is a voltage controlled circuit that allows control over polarity and voltage. The voltage is controlled by PWM to be able to get continuous control over the DC motor speed. PWM allows you to control the average input voltage by sending varying duty cycle pulse outputs from a digital pin. The polarity is controlled by flipping which way current flows through the H-bridge. For further details please review 14.

8.5 Discussion of Motor Control Library

The axis is controlled by using the Position object. The main methods we have written are discussed below. Much safety requirements can be added.

- **Position** This function is called when making a postion object.
- **StartPos()** sets the limit pins and enables the motor pins to as outputs and the voltage varying enable pin.
- **Calibrate()** this is the function that times how long it takes to get from one side to the other and gives it a base point to build off.
- **setMotor()** this should be a private function, but it was used for debugging. It manually sets the direction of the and voltage input to the transistors.
- **goTo()** allows the user to specify a percentage from the motor for the head to move to.

Further extensions:

- Interrupt to check if the head has drifted into the side.
- Exception handling to ensure the user input is not out of bounds.
- Automatic re calibration.
- Further improvement to the control accuracy. Some empirical measurements could be made to do this and a subsequent look-up table could be used.

```
#include "MotorControl.h"
#include <inttypes.h>
#include <Arduino.h>
#include <Button.h>
#include "ezButton.h"

#define M1p 11
#define M1n 8
#define M1e 9

// Functions in this file are for:
// - Calibrating the motor
// - Sending commands to the axis
// - This code is designed for detecting button presses
// - Only controls one motor
// __|                    |
// (+)-+|================|
// --|                    |
// Motor               Far Side
//But 3                But 2
// 0%                  100%
// Reverse = 1 means ---> going towards the motor
Position::Position(uint8_t motorSidePin, uint8_t farSidePin)
{
        constructPosition(motorSidePin, farSidePin);
}

void Position::constructPosition(uint8_t motorSidePin, uint8_t farSidePin)
{
        _motorSidePin = motorSidePin;
        _farSidePin = farSidePin;
}

void Position::StartPos()
{

        pinMode(M1p, 1);   // Input m1+ pin
        pinMode(M1n, 1);   // Input m1- pin
        pinMode(M1e, 1); // Enable m1

}

void Position::Calibrate()
{
                pinMode(M1p, 1);   // Input m1+ pin
                pinMode(M1n, 1);   // Input m1- pin
                pinMode(M1e, 1); // Enable m1


                ezButton limitSwitchFar(2);
                ezButton limitSwitchMotor(3);

                limitSwitchFar.setDebounceTime(20);
```

```
		limitSwitchMotor.setDebounceTime(20);


		setMotor(150, 1);
		while (1)
		{
			limitSwitchFar.loop();
			limitSwitchMotor.loop();

			if (limitSwitchMotor.isPressed()){
				Serial.println("The limit switch: UNTOUCHED -> TOUCHED");

				setMotor(0, 0);
				break;
			}

		}

		Serial.println("Reached Motor");

		delay(500);
		setMotor(150, 0);
		Serial.println("Calibrating");

		long t1 = static_cast<long>(millis());
		Serial.print("Start time: ");
		Serial.println(t1);
		while (1)
		{
			limitSwitchFar.loop();
			limitSwitchMotor.loop();

			if(limitSwitchFar.isPressed())
				// Serial.println("The limit switch: UNTOUCHED -> TOUCHED");

			if (limitSwitchFar.isPressed()){

			setMotor(0, 1);
					break;
			}
		}
		_currentPos = 100;

		long t2 = static_cast<long>(millis()); // End time (corrected for stoptime)
		Serial.println(t2);
		_travelTime = (t2 - t1); // Approximate time to move from one side to another

}


bool intFlip(int start, int final)
{
	int retVal;
	if (start>final)
```

```
        {
                retVal = 1;
        }
        else{
                retVal = 0;
        }
        return retVal;
}

void Position::goTo(long FinalPos)
{
        // This fuction goes to a specific point on the axis
        // currentpos
        long a = FinalPos;
        long LoopRunTime =  abs(((_currentPos - FinalPos)*_travelTime))/100;
        long end_time = LoopRunTime + static_cast<float>( millis());
        bool polarity = intFlip(_currentPos, FinalPos);


        int currTime = millis();
        while ( static_cast<long>(millis()) < end_time+200)
        {
                if (!(millis()%3000)){
                        delay(1);
                        Serial.print("Current Time: ");
                        Serial.println(millis());
                }
                setMotor(150, polarity);
        }
        setMotor(0,0);
        Serial.print("After goTo Current Pos: ");
        Serial.println(_currentPos);
        int t2 = millis();
        _currentPos = FinalPos;
        setMotor(0, 0);
}


void Position::setMotor(uint8_t speed, boolean reverse) // Set the polar
{
        analogWrite(M1e, speed);
        digitalWrite(M1p, reverse);
        digitalWrite(M1n, !reverse);
}
```

Listing 1: Motor Driver Code

9 Constraints of Design

9.1 Core constraints

This section clarifies the core constraints of the design and explains how the design will overcome or follow said restrictions or to produce a functional and quality product

- Efficient and Accurate Planting of Seeds
- Reasonable Product Pricing
- Use of Clear Agar Solution
- Accurate Vision Tracking

9.2 Efficient and Accurate Planting of Seeds

The primary reasons for the creation of the CN-Seeder is to increase efficiency in the laboratories and reduce costs of lab technicians having to manually plant the seeds. As the printer only has one seeder output, it can only print one seed per movement causing it to be time consuming to plant all the seeds. The CN-Seeder must be faster or close to the time it takes for a lab technician to plant the seeds to make it viable. Although the lab technician can work on another task while waiting for the seeder to print it is preferable to seed efficiently. The seeds are used to simulate crop growth and to get the most accurate results the CN-Seeder must accurately distribute the seeds at specified distances away form each other. There fore the motors must be accurate to prevent inconsistent seed placement which might ruin a lab test.

9.3 Reasonable Product Pricing

The CN-Seeder must be cheap to buy and maintain to incentives its purchase and make it viable compared to the lab technicians cost of planting the seeds individually. When interviewing Josh Mylne from Curtins Centre for Crop Disease Analysis, Mr Mylne states it takes him about 30 - 45 minutes to efficiently and correctly plant seeds into a standard petri dish. Using Curtin University Salaries [11] to estimate the pay for a lab researcher, mr Mylne is paid close to $102,087 per year. if working 40 hours a week for 50 weeks a year that's about $51 per hour costing Curtin roughly $25-$38 per petri dish planting (average cost is $31.5). The expected retail selling price is $3,173.22 according to the market research. This means the the lab technician must use the CN-Seeder 100 times to make it profitable. This means the CN-Seeder must work 100 times or more to make it a functional and worth while product.

9.4 Use of Clear Agar Solution

As there are sensors detecting the seeds flow, there cannot be any dirt or discolouration in the agar solution as it might ping a false true. The agar solution must be close to clear so the laser can pass through it to the receiver.

9.5 Accurate Vision Tracking for Plantation

As the CN-Seeder uses vision tracking to locate a pitri dish in its printing bed and plan its planting pattern for said dish. The accuracy must have a maximum of ± 2mm to be with in reasons else the seeds will be planted out side of the the pitri dish. The planting path must be editable on the embedded system for the lab technician to edit and customise it to follow specified placement locations.

10 Stretch Goals and Further Development

This report has comprehensively covered the initial development plan for the CN-SEEDER. As covered in the Work Breakdown Structure the initial design and construction of the device is scheduled to take a minimum of 51 business days, which represents a significant investment not only in time, but capital. With the initial launch and sale of our device we will then be able to gather data on its use from those within industry to add features and components which can be either of specific use to large clients, or to add additional functionality to the next generation model. The following features have been initially considered for inclusion in the second generation of printer.

10.1 Self-Contained Growth Environment

The most ambition of the proposed stretch goals is to create a self-contained growing environment that the experimenter is able to place the petri dishes within, and then leave for the duration of the experimentation and monitoring. This will encompass the addition of LED grow lights, temperature control, hydration, camera monitoring and time lapse photography.

10.1.1 Temperature Control

Although our initial design has sensors to monitor the overall temperature of our circuitry, there is currently no capacity for maintaining ambient temperature of the seeder itself, other than controlling the temperature of the room in which it is placed. To be able to create a growth environment for seeds the inclusion of LED Grow lights will also provide a heat source to increase the temperature of the contained space. A 15000 lux light source can be acquired for an additional cost of $52 [12]. This light already has integrated timing and can be set by the user for their desired intervals. This functionality can be integrated into our existing circuitry to be powered on or off depending on the pre-selected temperature or growth preferences to ensure that we do not raise the temperature above a certain threshold. To reduce the temperature the addition of a hydrodynamic fan to the case will provide air flow from the external environment to reduce the air temperature [13].

10.1.2 Seed Monitoring

An additional functionality that will at to the overall use and utility of our product is the inclusion of the hardware required to take detailed photography in a variety of light spectrums, as well as the software required to process, store and export this data to the user. The addition of a AS7265x Triad Spectroscopy Sensor would allow the user to take images and video in visible, Ultra Violet and Infrared light spectrums at an additional cost of $69.95 [14]. This will also require the addition of external storage which can be achieved by the addition of a 62GB USB directly into the Raspberry Pi USB port and the provision of code to automatically detect the external storage and save to it on a predetermined interval [15].

10.1.3 Automatic petri dish loading and unloading

An additional component of great interest to the team for future development is the functionality to be able to self-load multiple petri dishes onto the printing plate to be able to handle large batches of petri dishes and remove the requirement of constant monitoring, loading and unloading by the user. The proposed system will entail a Petri dish stand which the user will load the pre-filled petri dishes into at the rear of the machine. This will have an actuator connected at the base with a 200mm stroke length that will push the bottom petri dish onto the base, which will be modified to a conveyer belt. The tower itself will be connected to a belt driven linear slide which will move the tower along to load the print tray before the conveyer belt will move the entire base forward. This will enable several dishes to be loaded into the print base rather than having to print single dishes at a time. The 12V linear actuator has been costed at $50.34 [16]. The belt driven linear slide will be an additional $140.61 [17]. The conveyer belt itself will consist of a 600mm x 1400mm rubber sheet connected to 2 rollers and a 12 V motor, however the pricing of this will be dependent on the construction and component requirements, which will require further research and development, however initial estimates cost this at approximately $250.

10.2 Internet Connectivity

The proposed Raspberry Pi 4 component has an integrated BCM58712 chip which allows for network connectivity, however within the first iteration no network features have been integrated or utilised [1]. By implementing network connectivity, once the user has enabled a wireless or wired network on their device we will be able to push updates to their device, as well as be able to create push notifications that could be enabled to send to the users mobile or email to notify them of the current print cycle status, if the printer is running low on agar solution, if there are no more petri dishes remaining in the proposed loader, or if the environment has become too hot or cold. This additional functionality would also enable us to wirelessly transmit data from the storage device connected in the printer to the user on pre-determined schedules. This will however also bring the added complexity of having to ensure network and data security and as such was omitted from the initial build as a consultant or expert would have to be employed to ensure that we do not leave the user vulnerable to any malfeasance.

10.3 Self-Sterilisation and Syringe Flushing capacity

To ensure that no unwanted contaminants enter the environment, and to reduce the likelihood of bacterial build up within any of the internal components a self-sterilisation cycle and ability to flush both the internal tubing and syringe will be a priority for future development. This will necessitate the inclusion of an overfill or flushing bin to be integrated into the design as well as a system of either volumetric recording or sensing to notify the user after a predetermined number of uses that a flushing cycle is required to reduce or remove the possibility of contamination. The integration of either an indicator light, a notification on the LCD, a push notification to a device or a scheduled maintenance cycle will be necessary to ensure that the self-sterilisation occurs at required intervals, or programming the cycle into the menu items will be a useful addition to our device, and should increase the useful lifespan of the internal components before they are required to be replaced.

11 Maintainability

11.1 Software Maintainability Features

Below are some software features considered to improve product life cycle.

- **Debug Console:** having a debug console will allow for detailed information of information such as being able to access the machine's coordinate systems calibration, allow for testing communication between the gantry and the external hardware
- **Firmware updates:** the device will be able to get firmware updates through usb and over wireless.
- **Open Sourcing:** after being on the market for 5 years the source code will become available.

11.2 Frame Maintainability

The frame is composed of extruded aluminium section, attached with standard threaded hardware that fits into slots in the sections. The design from a maintainability perspective must be both strong enough to handle the actuation of the machine without shifting and putting the device out of alignment, modifiable in keeping with the modular design philosophy and allowing transformation of the device to suit specialized laboratory needs, and able to be assembled and disassembled with tools native to a (non-machinist) workshop or laboratory.

- The extruded surface of the section gives a semi-rough finish, meaning sections will have high static friction and be less likely to slide or move out of place during vibration in machine operation.
- High tensile strength machine screws will be used to fasten the aluminium sections with premade steel brackets. These screws will be torqued to specification on handover to customer, and a datasheet specifying torque specs will come with it for maintenance purposes.

- Square base section will have four screw-adjustable rubberised feet, both ensuring a flat and level operating surface and lessening the impact of a moving/vibrating surface on the fasteners.
- Threadlocker should be used both on the threaded bolts of the fasteners, and sparingly in the slotted sections where a fastener is attached.

11.3 Cleaning, Sanitization, and Refilling of Seeder System

Due to the organic nature of both the seeds themselves and the agar solution they are suspended in, it is vital that every component in contact with these consumables be easily and quickly removable, cleanable, and if needed replaceable.
Toward these goals, the Seeder System has a number of design features:

- The syringe and plunger tip are to be made of either medical or food-grade plastic and clearly labeled with a sterilization cycle number (amount of times the device can be cleaned/sterilized and reused, before degradation of the materials surface occurs and allows pathogens to grow). Luckily, much research into reducing medical waste has already explored this problem, and these figures can be drawn from existing studies.
- Considering the highly sensitive nature of some potential projects concerning *arabidopsis thalania* and other small flowering plants, some users may need to dispose of these components after a single use to ensure complete sterilization. The syringe and plunger (excluding the rack-and-pinion feed system behind the plunger) are of a standard size, and can easily be purchased pre-sterilized in bulk from a medical supplier. Fitting the sterilized syringes to the device via a slot and metal clip (shown in **Hardware Implementation - Feeder Seeder**) is a quick and easy process. Although the sale of the syringe and plunger is a potential market share that would be missed, the increased usability in delicate research make it a worthwhile trade off.
- Another benefit of standard swappable syringes is that many can be prefilled in advance and simply swapped in when one is emptied. The concept of a larger tank with attached pump to refill automatically was considered, but problems with pumping a viscous and heterogenous mixture of seeds and agar solution, as well as the obvious sanitary issues it would introduce made the concept untenable.

11.4 Motor and Drive Maintainability

Although far from the most sensitive or failure-prone area of the device, the motors and drive mechanisms should still be maintainable if the need arises. Design decisions ensuring a smooth maintenance process include:

- Standard length, width, and tooth spacing on all drive belts.
- Unshielded routing of belts between the motor and driven component. The visual inspection of and ease of access to the belts ensures the end user is aware of belt condition (on at least a basic, visual level) and can easily change them out if the need arises.
- All stepper motors will be attached to the device through a standardized mounting plate, affixed to it with metric bolts. They will also have a short lead and multi-pin connector clip allowing the motor to be detached without soldering or detaching lengths of wiring from the device.
- The driving belt gears will be fastened to the motor's driveshaft via a slot and clip or a tightening screw through the bore hole. Avoiding press-fitting the gears to the motor means the user doesn't need to purchase them specially or as a package deal.

12 Safety and Security Features

12.1 Electronic Safety

As our design employs electronics that can pull significant amount of electrical energy and are software controlled if this was to go to market we would need to ensure we follow AS/NZS 3000 wiring installation, 3008 cable selection, 4836 low power electronics and 3017 electronics inspection. This prototype can draw up to 1A at 17V and is potentially dangerous. In going to market the wiring will have to be properly insulated and designed. Further safety features that could be added are an ESD,

12.2 Biological Safety

The device will have to fitted to the specified and required physical containment requirements. Physical Containment 4 requires that equipment is:

- Operable by users wearing gloves
- No porous surfaces
- Must be designed to be easily disposed of during retirement
- Must have seamless edges for ease of cleaning
- Positive and negative pressuring inside of the device must be considereed

12.3 Physical Safety

Design Considerations in final design:

- Sound reduction.
- Finger guard rails to make it more work safe.
- Signage that indicates to wear eye glasses should be on the device.
- Should be designed to minimise strain.

13 References

[1] R. P. Ltd, *Buy a Raspberry Pi 4 Model B*, https://www.raspberrypi.com/products/raspberry-pi-4-model-b/.

[2] *Arduino Uno Rev3*, https://store-usa.arduino.cc/products/arduino-uno-rev3.

[3] *5MP Camera for Raspberry Pi*, https://raspberry.piaustralia.com.au/products/5mp-camera-for-raspberry-pi.

[4] Altronics, *Z6343 Funduino L298 H-Bridge Motor Shield For Arduino*, https://www.altronics.com.au/p/z6343-l298-h-bridge-motor-shield/.

[5] *SPDT 250V 3A Miniature Micro Switch with Lever | Jaycar Electronics*, https://www.jaycar.com.au/spdt-250v-3a-miniature-micro-switch-with-lever/p/SM1038.

[6] *J0016 48:1 Geared Motor With Plastic Housing*, https://www.altronics.com.au/p/j0016-48-1-geared-motor/.

[7] *RS PRO Through Beam Photoelectric Sensor, Barrel Sensor, 20 m Detection Range | RS*, https://au.rs-online.com/web/p/photoelectric-sensors/2043998?cm_mmc=AU-PLA-DS3A-_-google-_-PLA_AU_EN_Automation _-(AU:Whoop!)+Photoelectric+Sensors-_-2043998&matchtype=&aud-842313741498:pla-307534486281&gclid=CjwK ncVfjL4NFrvVhir-L67OBpDpPKYBoCRZoQAvD_BwE&gclsrc=aw.ds.

[8] *T-Slot Photo Interrupter*, https://core-electronics.com.au/t-slot-photo-interrupter-with-1-meter-cable.html, Wholesale Supplier.

[9] *Frame Rail 6063 Alu*, https://au-product.item24.com/en/productdetails/products/line-5-construction-profiles-1001042368/profile-5-20x20-natural-37003/, Wholesale Supplier.

[10] *Stepper Motor*, https://core-electronics.com.au/stepper-motor-with-cable.html.

[11] *Curtin University salaries in Australia: How much does Curtin University pay? | Indeed.com*, https://au.indeed.com/cmp University/salaries.

[12] *Amazon.com: LED Grow Light for Indoor Plants, Relassy 15000Lux Sunlike Full Spectrum Grow Lamp, 45W Auto ON/Off , 3/6/12H Timing Function, 4 Brightness Levels, Professional for Seedling Growing Blooming Fruiting : Patio, Lawn & Garden*, https://www.amazon.com/gp/product/B07C68N7PC/ref=as_li_tl?ie=UT 20.

[13] *80mm Silent Hydrodynamic Bearing Case Fan | Jaycar Electronics*, https://www.jaycar.com.au/80mm-silent-hydrodynamic-bearing-case-fan/p/YX2570?gclid=Cj0KCQjw48OaBhDWARIsAMd966CJ3zhCaKuEoSo5qPy-NGtX_N_QC5UGWQfeuKmlmAzFwXeB64pXCp4aApotEALw_wcB.

[14] *SparkFun Triad Spectroscopy Sensor - AS7265x (Qwiic) - SEN-15050 - SparkFun Electronics*, https://www.sparkfun.com

[15] *Buy Now | SanDisk Cruzer Snap 64GB USB2.0 Flash Drive | PLE Computers*, https://www.ple.com.au/Products/64325 cruzer-snap-64gb-usb20-flash-drive?gclid=CjwKCAjw2OiaBhBSEiwAh2ZSP53WI3Kc1cXSaN0Y30cXXaspVzBg6sTkN

[16] *1500N/330lbs 12V Linear Actuator Electric Motor for Auto Car Window Door Open IG*, https://www.ebay.com.au/itm/

[17] *1000mm Belt Actuator Kit*.