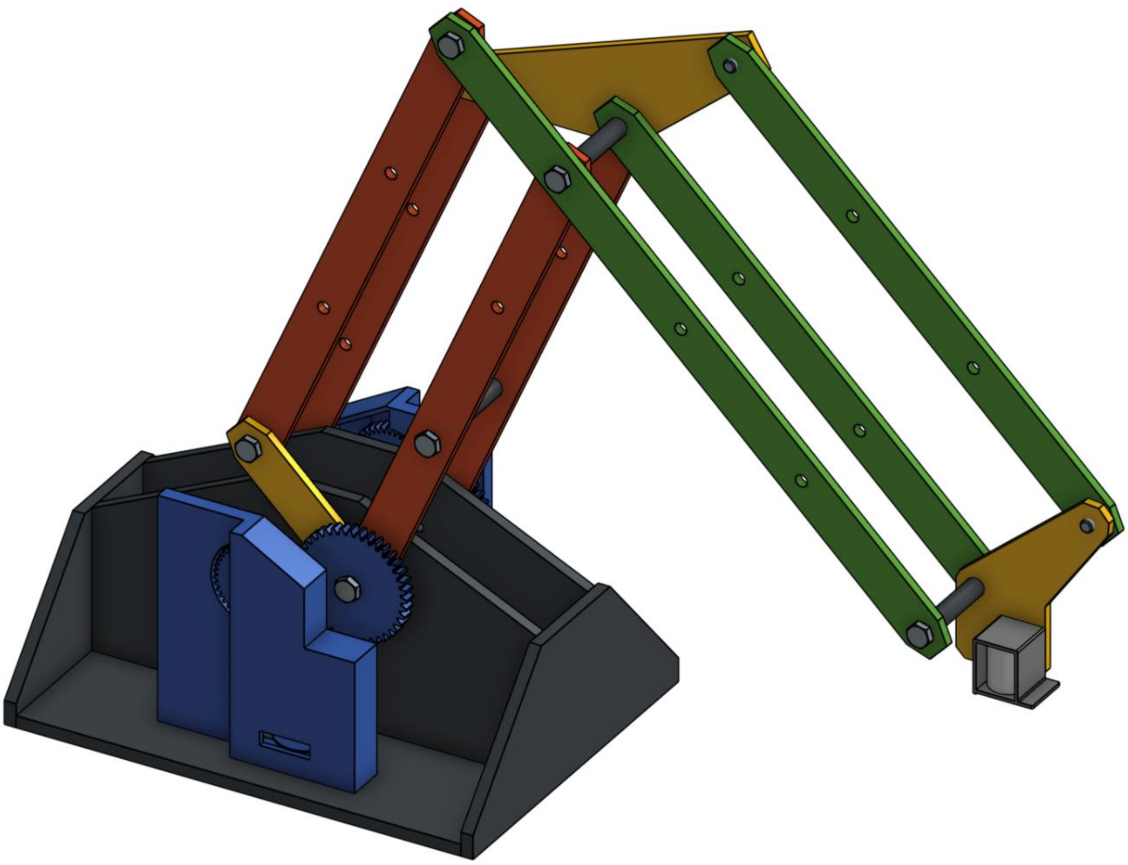
RUBBISH ROBOTICS

Robot arm manufacturing instruction manual



2 Degree of freedom 4-bar mechanism robotic arm

Instruction manual

Not Rubbish robotics, The University of Edinburgh, Edinburgh, United Kingdom

First Edition, Not Rubbish Robotics, Edinburgh, United Kingdom/2023

1. Introduction

Welcome to the NRR robot arm instruction manual!

Here we hope to outline how you can make your very own robot arm from nothing more than just a printer, an old laptop charger, some hand tools, and optionally an Arduino and some motor control boards.
This is intended to equip you with the information and abilities to construct your very own e-waste robots, whether you are an experienced enthusiast or a curious beginner.

From the Not Rubbish Robotics team, we wish you the best of luck!

We would love to see pictures of the robots you have built using our kit! Please reach out to notrubbishrobotics@gmail.com.

Table of Contents

WARNING!

Before attempting to disassemble any printers and use their components to build a robot arm. Please read the following section very carefully. And if unsure at any stage please consult your instructor if available. Mishandling power tools or electronics could result in severe injury.

Any steps you take from here on are at your own risk and Not Rubbish Robotics is not liable for any injury.

2. Risk Assessment

It should be noted that as we are dealing with e-waste. The devices being deconstructed are likely to be damaged in some form, either due to damage that caused them to be decommissioned or otherwise due to transport to a waste disposal facility. Disassembling E-waste (in this case printers) involves working with various components and potential hazards. It is crucial to prioritise safety throughout the process. The following table outlines key hazards and corresponding mitigation strategies of components to ensure a safe disassembly procedure.

Components	Hazards	Likelihood	Severity	Mitigation Measures
Circuit Boards	Sharp edges, potential electric shock (capacitor)	Low	High	Handle boards with care and from edge to avoid cuts/electrocution, and use anti-static precautions to prevent damage
Capacitors	Potential electric shock, potential chemical leakage	Low	High	Wear insulating gloves and use an insulating screwdriver. Discharge the capacitor and watch for leakage
Wiring/cables	Sharp edges and damages	Medium	Medium	Wear gloves and handle with care to prevent any fraying/bending
Print Head	Possibility of ink spillage and exposure to printhead elements	Low	Medium	Wear gloves and handle with care to avoid ink exposure
Batteries	Risk of exposure to corrosive materials and potential leakage	Low	Medium	Identify/remove with proper tools and place in a designated hazardous waste container if damaged
Plastic Parts	Potential sharp edges if broken, risk of cuts	Low	Low	Wear gloves and handle with care to avoid any breakage.
Metal Parts	Rusting/sharp edges with risk of cuts	Low	Low	Wear gloves and use proper tools for handling metal components

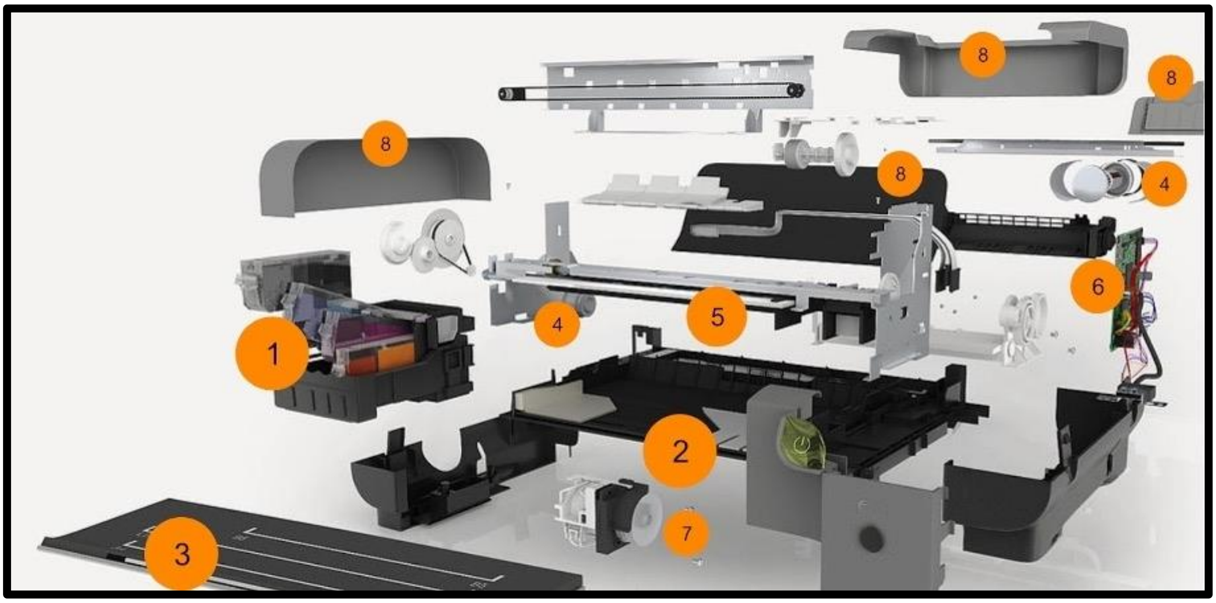
Glass sheet	Potential sharp edges if broken, risk of cuts	Low	Medium	Wear gloves and handle with care to avoid any breakage
Ink Cartridges/ Toners	Ink exposure and chemical inhalation	Low	Medium	Wear a mask and gloves. Work in a well-ventilated area
Screws/ Fasteners	Small parts, potential choking hazard	Low	Medium	Keep them in a designated container, keep work area organised
Plastic particles	Potential inhalation of particulars when sanding/filing	Low	Low	Wear a mask or ventilate the area if available

3. Printer Disassembly

3.1 Printer parts and rough location

Before beginning the disassembly procedure, it is important to make yourself familiar with the various components of the printer. This preliminary step will help you navigate through the internal structure with confidence and speed up the process. The table below presents the key components frequently found in printers, followed by an exploded view of a typical printer.

Component	Description	Location	
Print head	Contains nozzles and ink chambers	Near the cartridge slots	(1)
Paper tray	Where paper is loaded from	Bottom, front of the printer	(2)
Paper feeder	Guides paper into the printing area	Above Paper Tray	(3)
Motors	Mechanically drive the printer movements	Near Gears	(4)
Rollers	Rubber/plastic components that help feed paper through the printer	Along the Paper Path, paper tray, fuser unit	(5)
Circuit board	Electronic components that control the printer functions	Near Control Panel	(6)
Fuser unit	Heats/Bonds toner to paper	Exit of the print path	(7)
Housing/Casing	Protects internal components	The outer shell of the printer	(8)



3.2: Disassembly

You should aim to source the following parts from the printers you disassemble for the kit. These are:

- Motors
- Any flat plastic sheets
- Gears or pulleys
- Electromagnet
- Screws and bolts
- Wires
- Buttons and switches

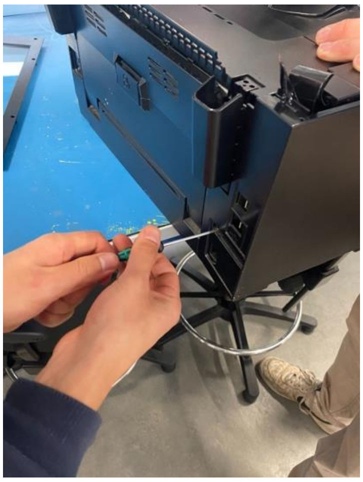
You can now begin the disassembly process of your printer. Please ensure you have properly read and understood the safety section of this manual.

Step 1: Unplug and remove the power supply:

Sub-step 1.1: The power supply unit is usually a separate component, often located at the back of the printer. Disconnect it and remove any screws holding it in place.

Step 2: Open the printer casing

Sub-step 2.1: The outer casing of the printer is typically held together by screws or clips. These are commonly found on the sides, back, and sometimes underneath the printer. Removing the casing allows access to the internal components. Be cautious not to damage plastic tabs or connectors when opening it as these parts will be valuable for the construction of the robot.



Step 3: Remove ink cartridges - inkjet printers (wear gloves or have tissue at hand)

Sub-step 3.1: Open the printer cover: Lift or open the printer cover. This is the part of the printer where the print head and ink cartridges are located. It's usually easy to identify and access, as it's designed for regular cartridge replacement.

Sub-step 3.2: Wait for the carriage to move: After opening the printer cover, the printer's carriage, which holds the ink cartridges, may move to the centre of the printer. This is a standard safety feature to provide better access for cartridge replacement

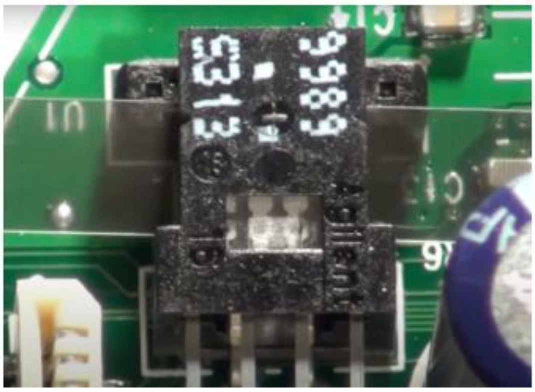
Sub-step 3.3: Identify the cartridges: The ink cartridges are typically small, rectangular, or oval-shaped containers with colour labels (e.g., black, cyan, magenta, yellow). They are usually located on the carriage.

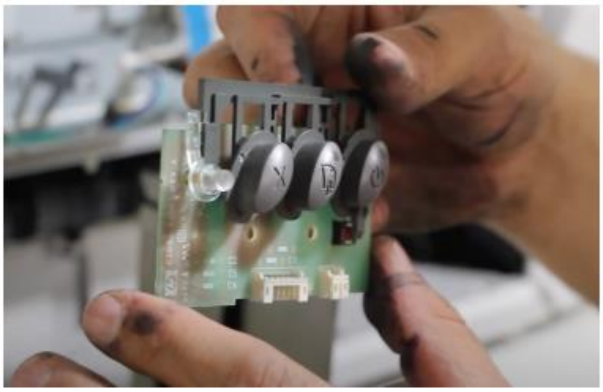
Sub-step 3.4: Remove the old cartridges: Gently push down on the cartridge to release it from its holder. The cartridges are often secured by a clip or a latch that you can release. Once released, you can carefully pull out the old cartridge.

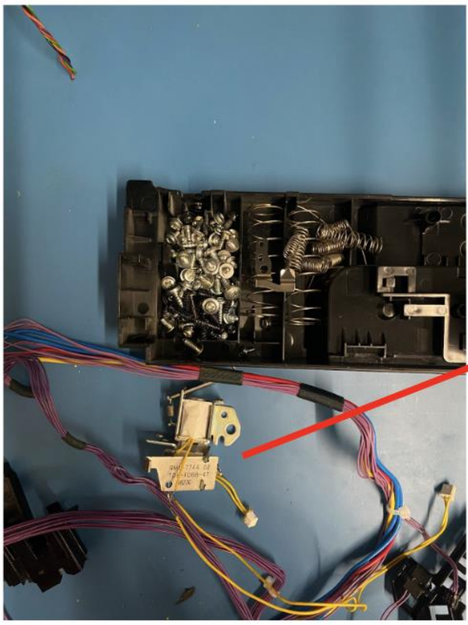
Step 4: Preparation

Hint: Throughout the following steps, take care to unscrew everything carefully and not pull-on cables very hard. The cables come with valuable attaching heads, which are very useful for future projects.

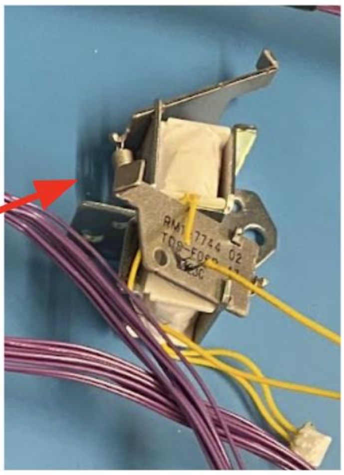
Step 4.1: Prepare a small container (which could be a part of the printer itself) to collect all screws and springs that can get easily lost. Also make sure to set apart any sensors, buttons, switches and electromagnets - some of which may look like this:



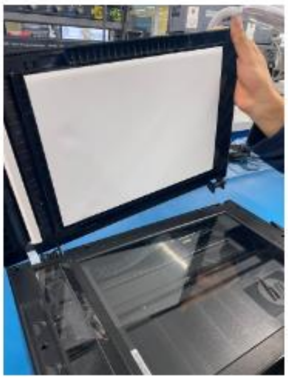


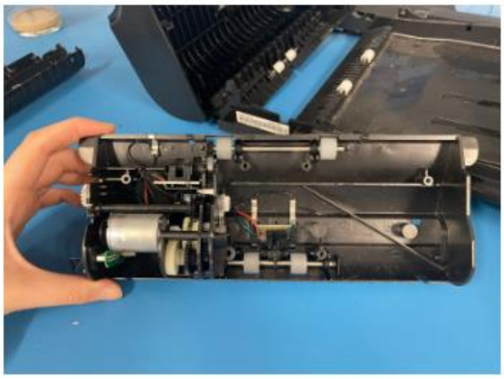


electromagnet



Step 5: Detach scanner assembly





Step 6: Locate motors from the scanner assembly (from the scanner lid)

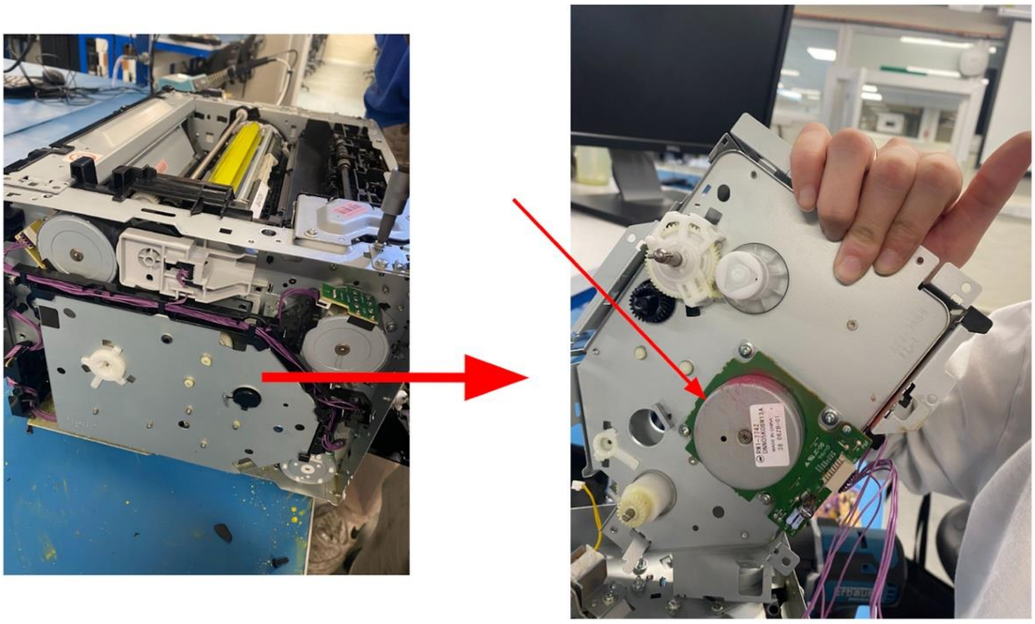
Step 7: Remove the external casing on the sides of the printer.

Hint: Motors, sensors, and springs are distributed throughout the printer, often near moving parts. These components may be hidden beneath plastic covers or near the paper path.

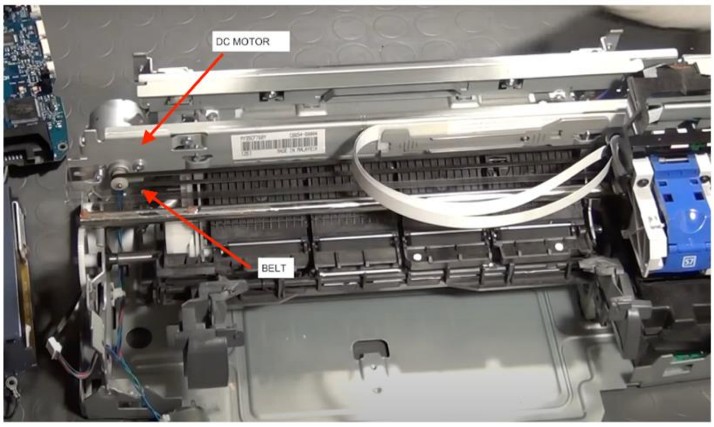
Step 7.1: Unscrew all motors and detach all cables.

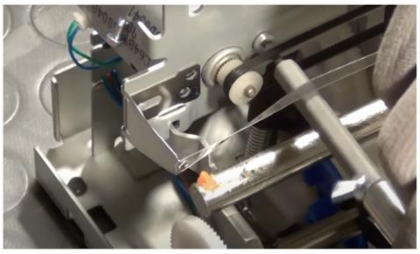


Step 7.2: Unscrew all covers, as they may reveal more motors inside. In this example, detaching the cover has revealed another motor.

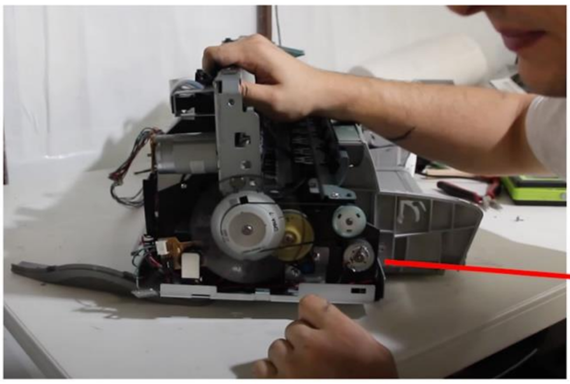
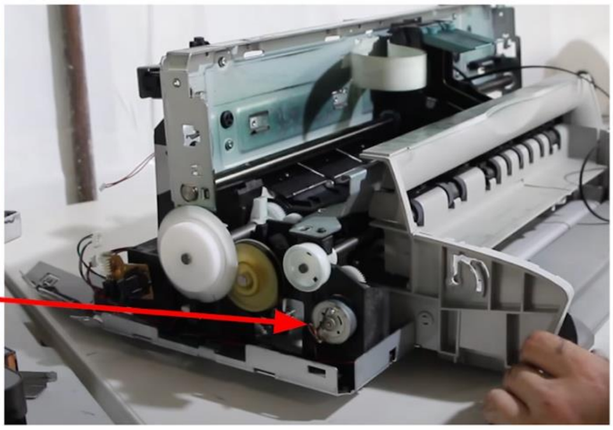


Step 7.3: Once all external panels are removed, you will be able to see the inner workings of the printer. Most printers will have a large DC motor that moves the printer head in the X direction. This is usually held in place by 2 or 3 screws, make sure to keep these screws. For example, loosely attach the screws back onto the motor straight after to make sure they stay with the motor. You should also keep the belt.





Step 7.4: With this same arrangement turned sideways, you will generally find another motor that works to feed the paper through the printer

Step 8: Axle removal

Step 8.1: The remaining axles can generally be loosened by turning a plastic lever at each end of the axles. Then they can simply be pulled out.

Step 8.1: Handle the axles with ink with care. If possible, wear gloves or handle them with tissue paper. Carefully put them to the side.

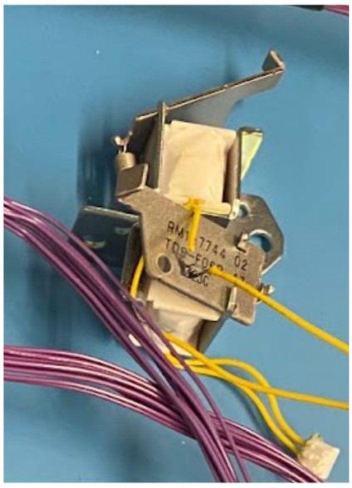
Hint: You will generally find a steel rod as part of the axle assembly, this is a high-quality component that can be reused for various mechanical applications and should be kept

Step 9: Extra screw removal

Step 9.1: Continue removing all the screws you can see in order to reveal any components specific to the printer you are handling.

Step 10: Important and required parts for the kit

Step 10.1: From this disassembly, the parts you will need to build the robot are: the motor assembly from the scanner, the electromagnets, and the flat plastic panels from the outer casing of the robot.

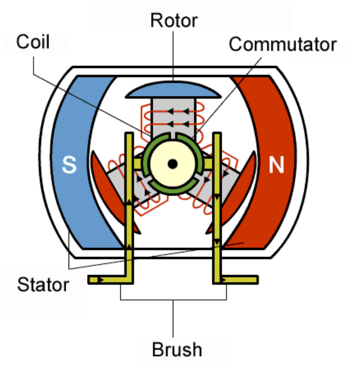


4 E-waste classification

4.1 Motor Identification

There are usually two commonly found motors within printers which are DC and stepper motors and differentiating between them is crucial in robotics, particularly when it comes to our robotic arm kit. DC motors enable continuous rotation, whereas stepper motors allow for precise control through incremental steps. This understanding allows us to effectively repurpose e-waste, guaranteeing the success of our kit. Below are descriptions of how to tell the difference between the two.

4.1.1 DC Motors

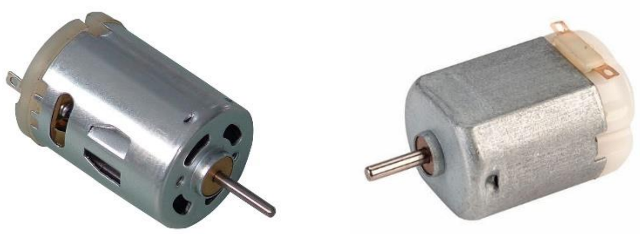


The **first type is a Brushed DC motor**. Brushed DC motors are relatively straightforward in design and operation. They typically consist of just two wires for power supply and can be integrated into a printed circuit board (PCB) by soldering them in place. These motors are commonly found in a wide range of electronic devices, including cordless tools, car window actuators, and blenders. However, they have seen a decline in use in recent years, being replaced by more efficient brushless DC motors.

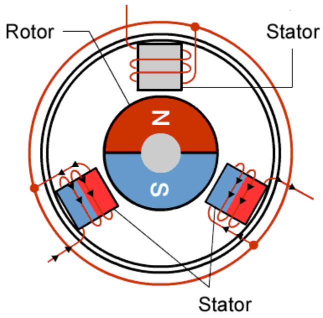
These motors consist of a stationary part (stator) and a rotating part (rotor). The rotor contains physical brushes and a commutator, which are vital components for their operation. The brushes make electrical contact with the commutator, allowing current to flow into the rotor windings, and creating a magnetic field. This magnetic field interacts with the stator's fixed magnets, generating the rotational motion.

These motors are user-friendly as they don't require a specialised controller for operation due to their uncomplicated structure. They run on direct current (DC) electricity. They exhibit wide voltage compatibility, with most small electronic devices utilising brushed DC motors operating within the 5 to 24V range.

However, there is a downside to their simplicity. The brushes in brushed DC motors tend to wear down over time as they constantly come into contact with the commutator. This wear and tear results in decreased motor lifespan and efficiency, making them less durable compared to brushless DC motors. Examples of such motors can be seen below.



The **second type is a Brushless DC motor.** Brushless DC motors are more advanced in their design, typically requiring at least three wires for operation due to their increased complexity when compared to brushed DC motors. They have gained significant popularity in recent years, largely replacing brushed motors due to their better longevity and higher energy efficiency. You can encounter brushless DC motors in a wide array of applications, including electric bicycles, hard drives, CD/DVD players, newer cordless tools, and various handheld battery-powered tools.



These motors feature a rotor and a stator, similar to brushed DC motors. However, they lack the brushes and commutators found in brushed motors. Instead, they rely on advanced electronic components, including Hall effect sensors, to determine the rotor's position. This eliminates the need for physical brushes and commutation, making brushless DC motors more efficient and long-lasting.

To set brushless DC motors in motion, they require a specialised controller. The controller precisely controls the current flow to the stator windings, ensuring smooth and efficient rotation. Examples can be seen below.

4.1.2 Stepper Motors

Stepper motors are a unique subset of brushless DC motors typically featuring a minimum of four wires for operation. What sets them apart is their ability to perform highly precise movements, which depend on the specific electrical input they receive. These motors find extensive use in applications where the exact rotation of the shaft is critical, such as in high-resolution printers, scanners, camera lenses, CNC machines, and 3D printers.

Stepper motors possess multiple coils within the stator. These coils are energised in a particular sequence to generate magnetic fields, and these fields interact with the rotor's magnets to induce movement. The key feature is that this movement occurs in discrete steps or increments. The precise control of these steps allows for accurate positioning and rotation, making stepper motors ideal for applications demanding precision.

To set stepper motors in motion, they require complex controllers and often microprocessors. These controllers send precise signals to the coils, directing them on when and how to energise, thus controlling the rotational steps. Due to their high torque characteristics, stepper motors typically require a higher current supply compared to other types of motors. Examples can be seen below

5. Robot Construction

5.1 Introduction

Welcome to the manufacturing section of the NRR kit. It's great that you have made it to this stage, this is where all of your work so far will start to take shape in the form of your very own controllable robot.

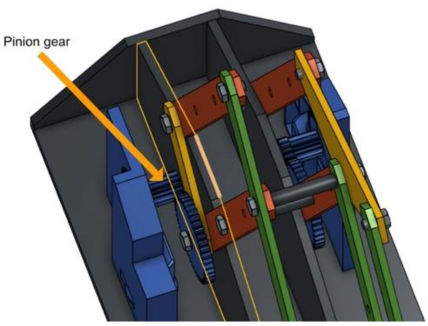
There will be three variations of our robot that you will be able to build with this kit. These variations differ by the type of motor that is used along with the controlling method of the robot. These three are:

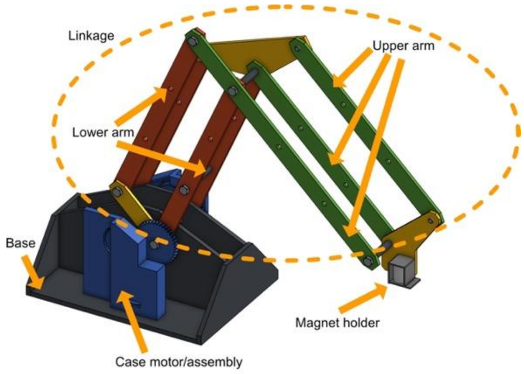
- The digitally controlled DC motor robot
- The digitally controlled stepper motor robot
- The analogue controlled DC motor robot

The benefits of these are outlined in section **3.2. Motor Identification.** Choose the motor type you want to work with and skip the sections which are only relevant to the other type.

You must keep in mind that we have tailored our instructions to make use of the most basic of resources which includes the most limited of tools. If you have access to more advanced machinery such as a belt sander or 3D printer, please feel empowered to use these to modify the design and create your own unique robot arm.

Below are images describing the nomenclature in the robot assembly which will be referred to continuously throughout the following section





5.2 Methodology

First of all, you are advised to familiarise yourself with the provided OnShape document. For example, try changing some of the design parameters and see how it affects the dimensions of the robot. Also, make sure you familiarise yourself with the nomenclature of the robot components as they will be referenced heavily throughout these instructions.

Next, you will be taking measurements of all your previously salvaged components from the printer. These measurements will then be used to generate a CAD model custom to your specific components.

Afterwards, you will print the templates from the CAD model that is generated for each individual part of the robot and send them to a working printer (I hope you haven't disassembled it already). These printed templates will then be attached to the flat plastic sheets that you have salvaged to create each separate component of the robot. These components will be cut out and attached together according to the CAD model with screws from the printer.

The motors will then be mounted to the base with gears connecting them to the arm, all salvaged from the printer. These motors will then be wired up to either an analogue circuit or your Arduino depending on the robot variation you have chosen. The electromagnet will also be attached to the end of the arm and further connected to the chosen controlling method. Lastly, if you chose the digitally controlled robot then you can code it to make it perform the desired operations.

5.2.1 Materials and Tools

The lists below display the materials and the tools that are required to make the kit come to life. All of these materials should have been found during the disassembly process of the printer which is pivotal in completing this kit. Any other optional components such as switches or LEDs can be added to the design for any desired extra functionality or character.

Essential e-waste components:

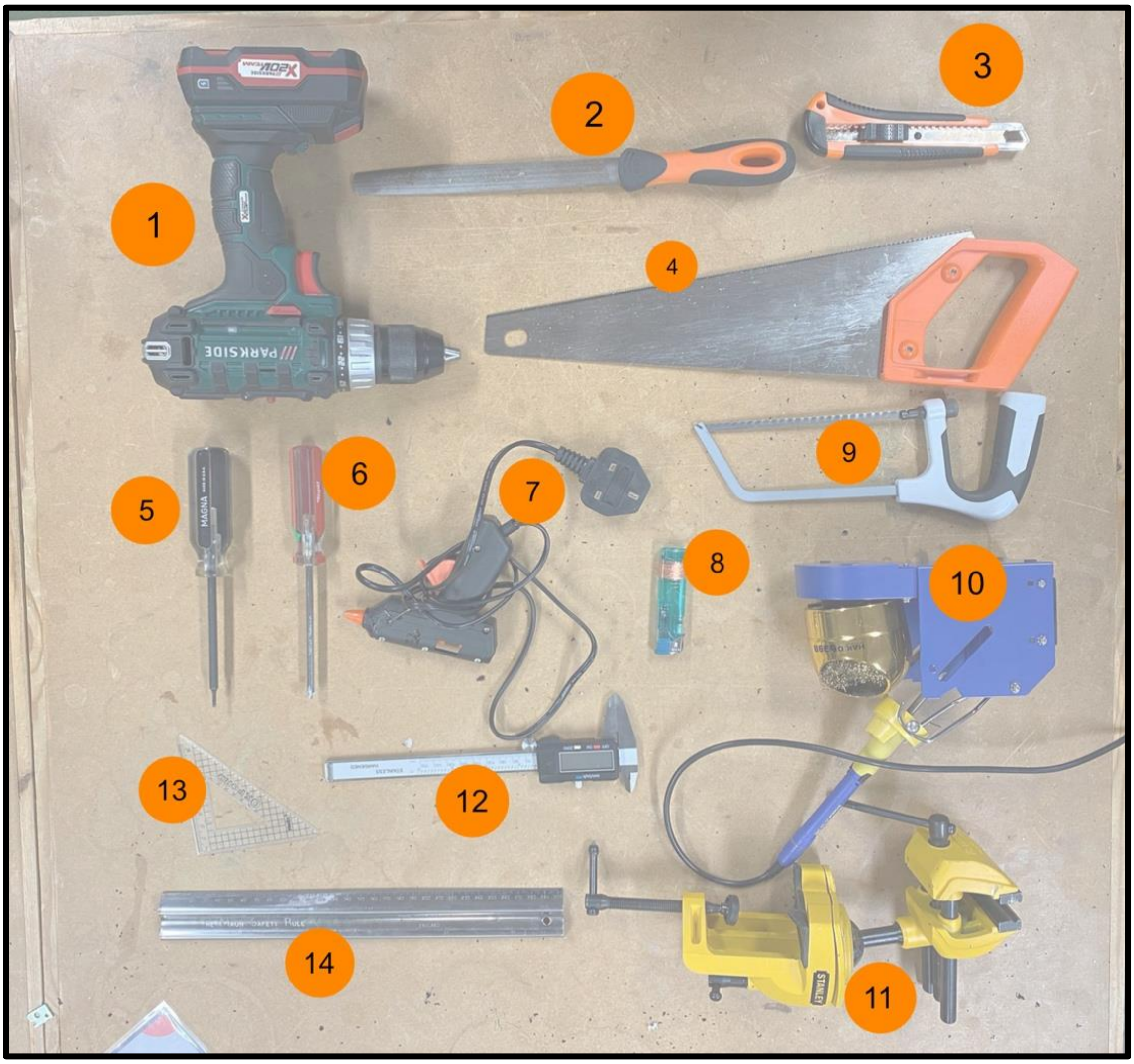
- Flat plastic sheets (most only need to be flat on one side)
- Screws, nuts, washers and bolts
- Gears
- Motors (can be stepper or DC)
- Electromagnet

Optional e-waste components:

- LEDs
- Switches

Required Tools:

- Power or Hand drill (with drill bits) (1)
- File (2)
- Scissors or Stanley Knife (3)
- Wood saw (4)
- Screwdriver set with torx (flat and Philips heads) (5&6)
- Glue gun (7)
- Lighter (8)
- Metal saw (9)
- Soldering Iron (10)
- Bench vice (not strictly necessary but helpful) (11)
- Callipers (optional but recommended) (12)
- 90 deg angle (13)
- Ruler (and preferably callipers) (14)



Required consumables:

- Hot glue (not essential but very useful when prototyping)
- Epoxy glue
- Sandpaper
- Pin connectors (you can use salvaged wires)
- Glue stick or general-purpose glue

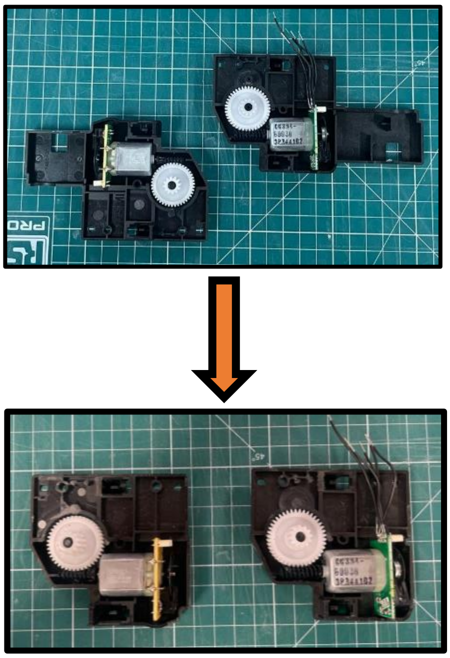
5.3 Manufacture

Step 1: Component measurements

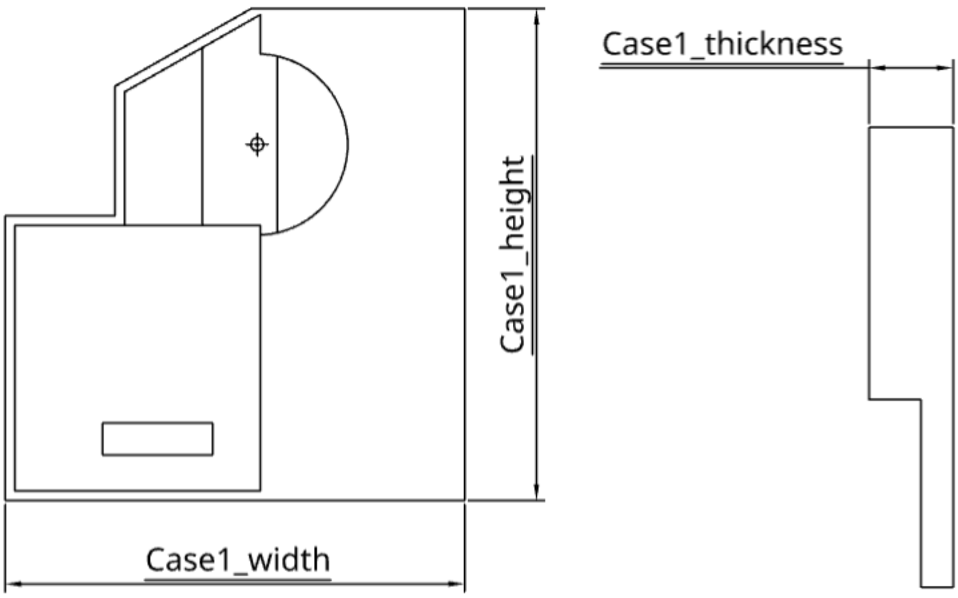
First and foremost are the measurements of the motors that you will be using in your robot, in this manual we will discuss two types of motors commonly found in printers. This is the scanner track motor assembly found in nearly all HP printers with a built-in scanner or copier. The round profile stepper motors are found in slightly larger printers. If you are using DC motors see step 1.1, otherwise, if you are using stepper motors see step 1.2.

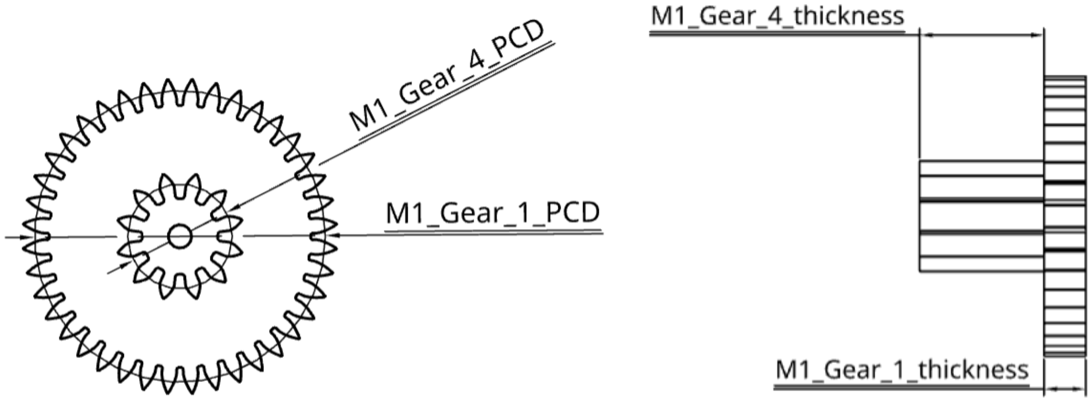
Step 1.1: DC case-motors

Step 1.1.1: If you have salvaged the scanner drive motors from an HP printer, then they should be like the below images. These cases must be modified to be used in the robot, using a wood or metal saw, make cuts to the case. Then use a file to ensure that all edges are perfectly flat, as these will be used to align the motors. The following images show before a before and after comparison of this.



Step 1.1.2: The measurements for the dimensions of one of the DC motor cases need to be taken and inputted into the Variable Studio table within the OnShape CAD. There are three measurements to take which are displayed below.

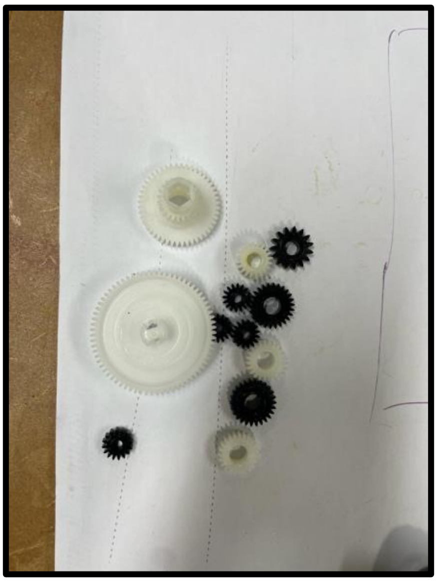




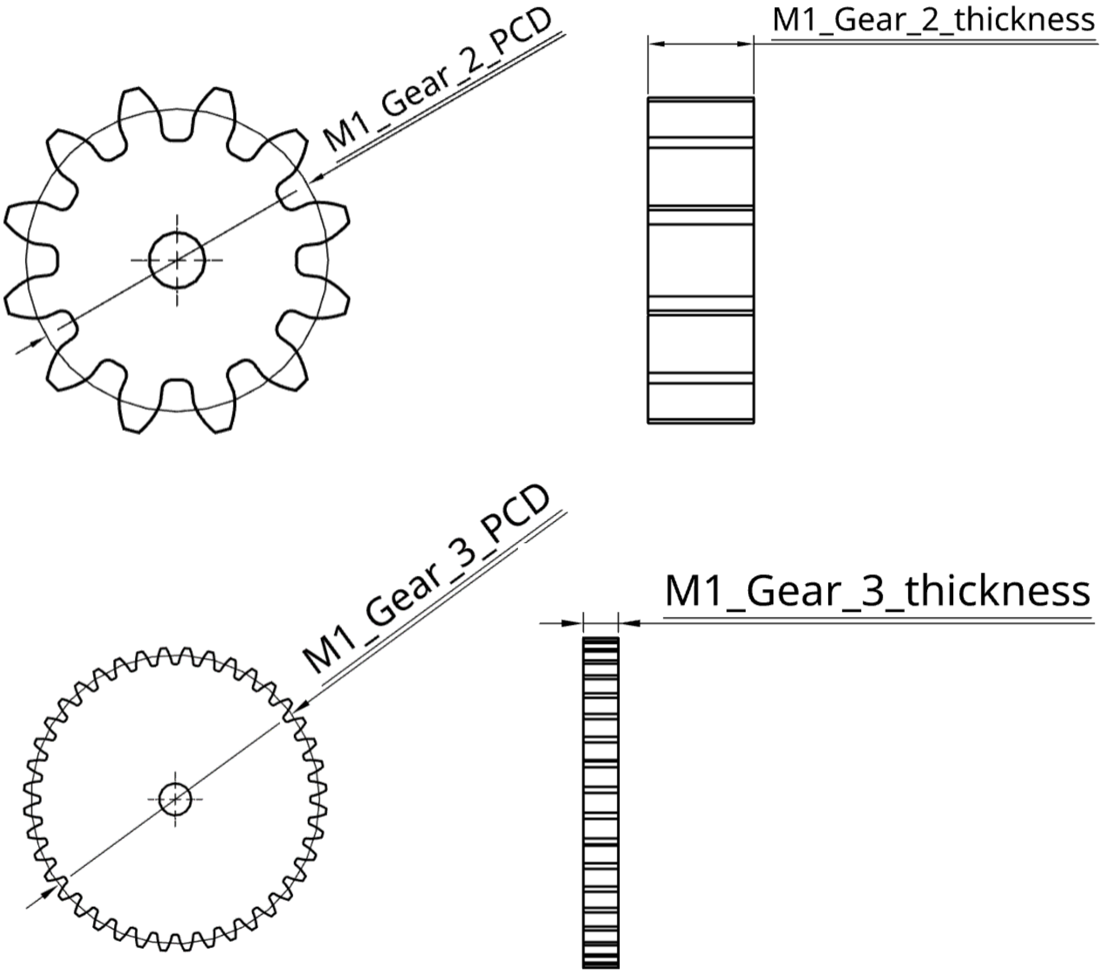
Step 1.1.3: Now gather all the gears you have salvaged from your printer. You will need two pairs of well-meshing small and large gears, ideally with a gear ratio of greater than 5. By counting the teeth and using the formula below, you can calculate the gear ratio.

$$Gear\ ratio = \frac{drive\ gear\ teath\ number}{pinion\ gear\ tooth\ number}$$

For our example, we used a 10 and 80 tooth pairing and a 16 and 80 tooth pairing which Gives an 8:1 and 6:1 gear ratio.

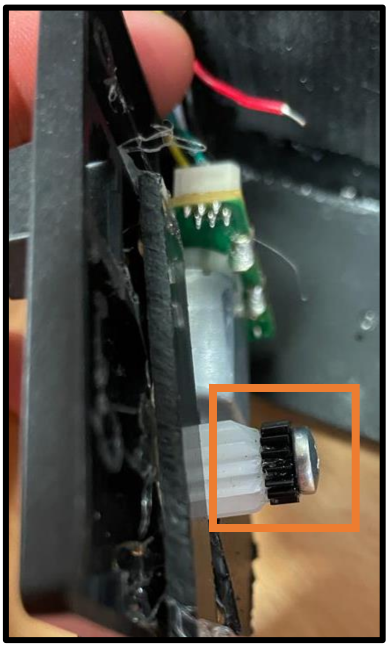


Step 1.1.4: Take one of the pairs of gears to use with one of the motors and take the following measurements. Input these measurements into Variable Studio table within the OnShape CAD.

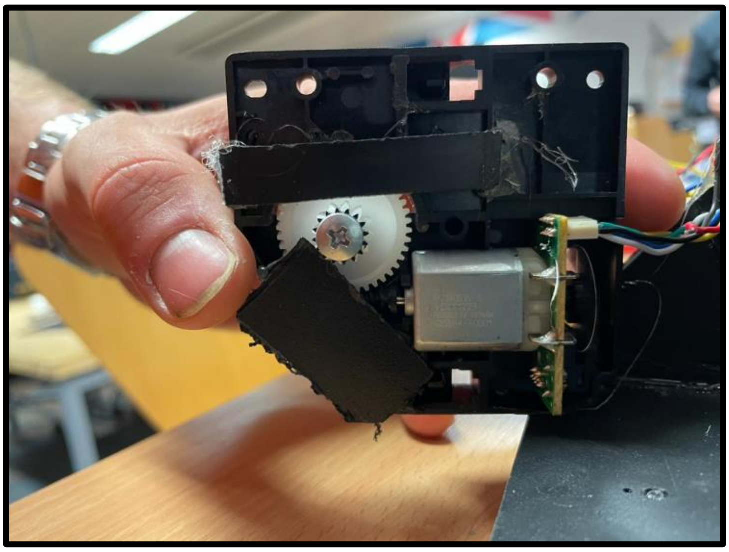


1.1.5: Now Gear 2 of needs sticking onto gear 4. To prepare the gears for gluing it is recommended to flatten the top surface of the output gear and then proceed to use a box cutter to scuff up both surfaces for better adhesion. Also, use some mineral spirit to remove any gear lubricant that might prevent good adhesion.

Then, mix up a small batch of epoxy and apply it to the gears while taking care not to get any epoxy into the teeth of the gears as this will cause bad meshing of the gears. It is important to make sure that the centre of the pinion gear is aligned with that of the output gear.



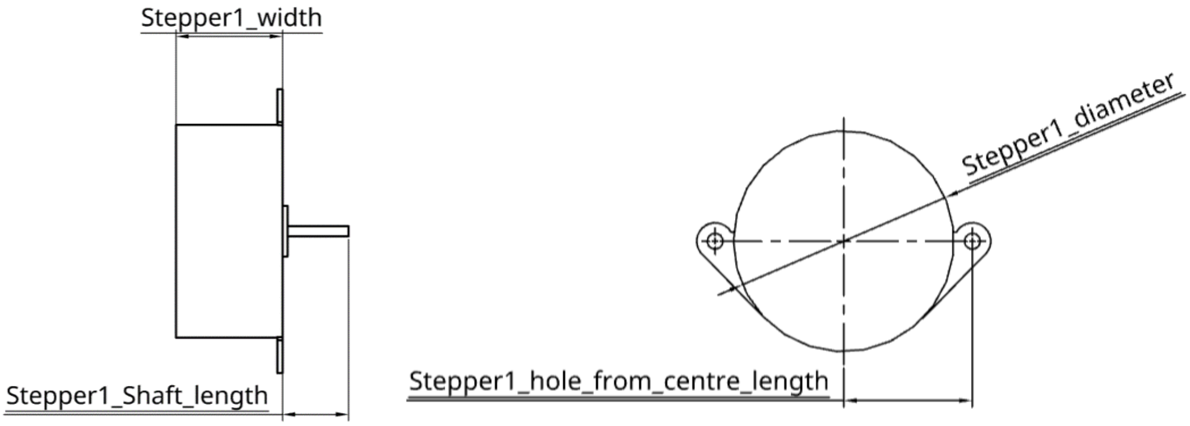
Step 1.1.6: The motor unit is no longer in the printer so the output gear can become dislodged from the assembly. to prevent this, take a short and thin scrap piece of plastic and glue it slightly overlapping the output gear. This should secure it in place. The gear and motor unit assembly can now be set aside to be used in subsequent stages.



Step 1.1.7: Now repeat **step 1.1.2** to **step 1.1.6** for the second DC motor case assembly. If both are done move on.

Step 1.2: Stepper motors.

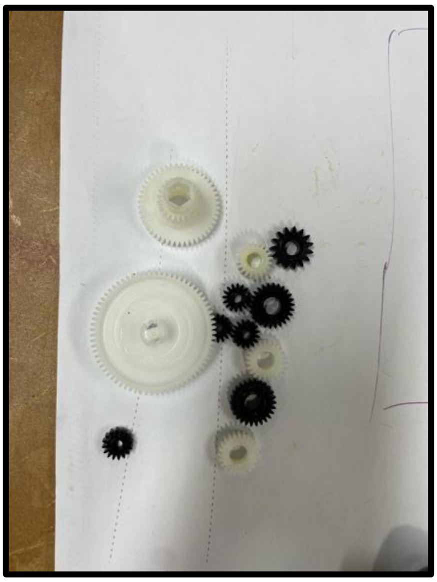
Step 1.2.1: The stepper motors you will possibly salvage are likely to take the following format of being a thick round disk. They are likely to have a metal helical gear mounted to their ends. Take the following measurements of the stepper outlined below and input these into the OnShape variable table.



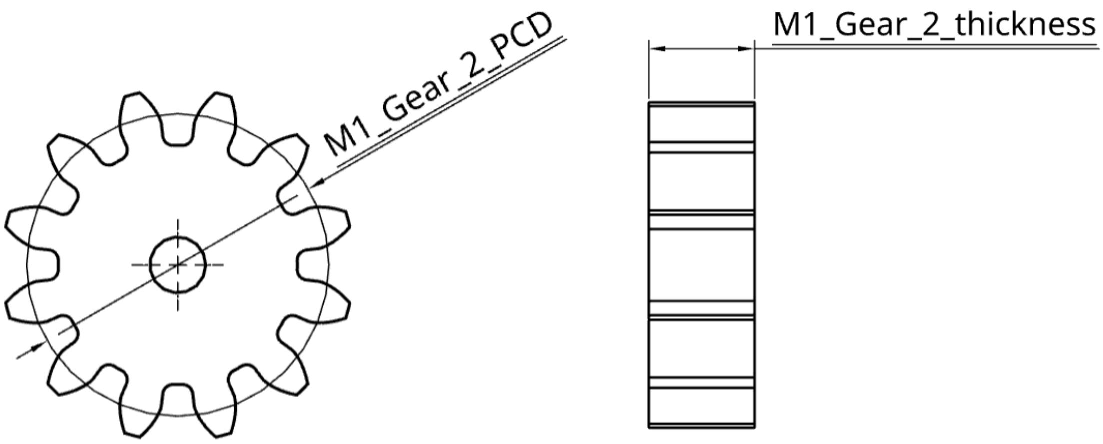
Step 1.2.2: Now gather all the gears you have salvaged from your printer. You will need two pairs of well-meshing small and large gears, ideally with a gear ratio of greater than 5. By counting the teeth and using the formula below, you can calculate the gear ratio.

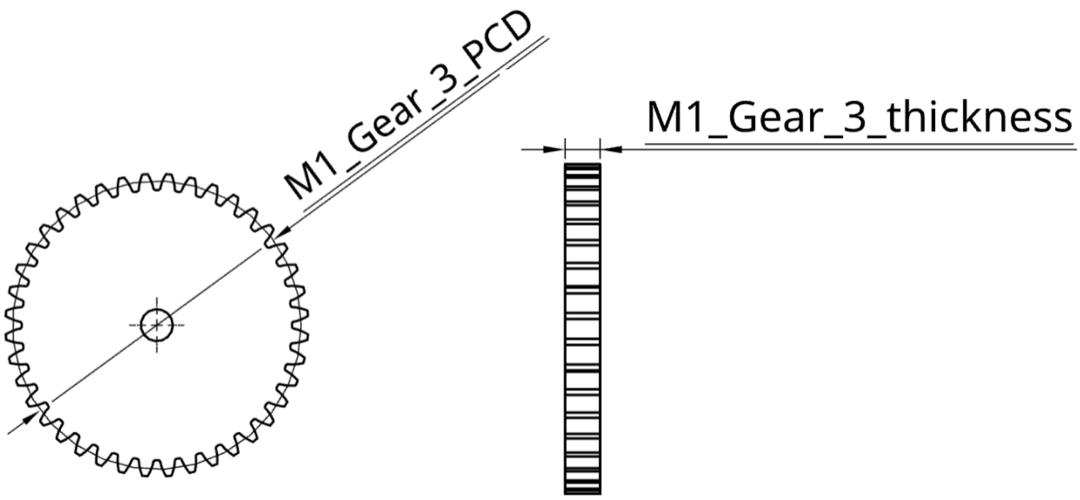
$$Gear\ ratio = \frac{drive\ gear\ teath\ number}{pinion\ gear\ tooth\ number}$$

For our example, we used a 10 and 80 tooth pairing and a 16 and 80 tooth pairing which Gives an 8:1 and 6:1 gear ratio. The smaller of the gears will be referred to as the pinion gear and the larger gear will be referred to as the drive gear.



Step 1.2.3: Take one of the pairs of gears to use with one of the motors and take the following measurements. Input these measurements into Variable Studio table within the OnShape CAD.





Step 1.2.3: Sometimes there is a metal gear attached to the end of the stepper motor shaft. If this is the case follow this step and then skip to **step 1.2.5**, if it is not the case then skip to **step 1.2.4.**

Now to mount the pinion gear. Firstly, drill a hole roughly 1-2 mm smaller than the outer diameters of the stepper motor shaft gears. Next, use a lighter to heat up the stepper output gears for approximately 30 seconds and press the pinion gear onto the stepper gear ensuring that they remain horizontal during the process. As the gears cool the pinion gear should be rigidly fixed in place.

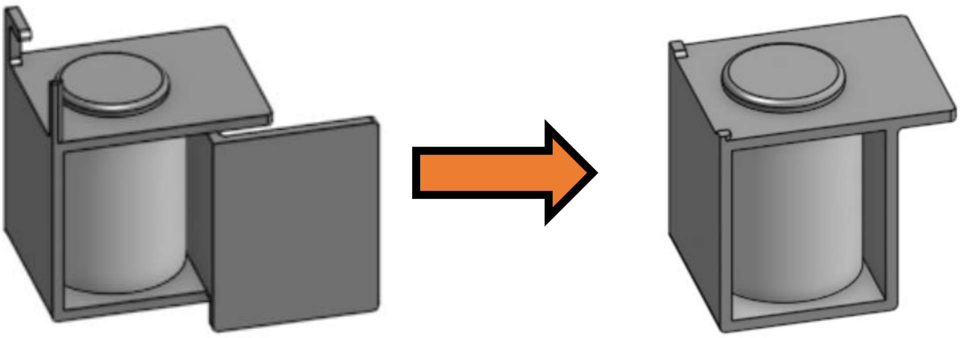
Step 1.2.4: Glue the pinion gear onto the end of the stepper motor shaft using a very small amount of epoxy.

Step 1.2.5: Now repeat **step 1.2.1** to **Step 1.2.4** for the second stepper motor**.** If both stepper motors are finished set aside to be used in subsequent steps.

Step 1.3: Electromagnet

The electromagnet that you have salvaged will likely take the appearance of the one we found in each of our HP printers being compact and lightweight.

Step 1.3.1: The electromagnet will most likely have additional metal around it. By using a metal saw, take off any of this extra metal. An example can be seen below.



Step 1.3.2: The electromagnet does not need measurements taken of it as this just sticks to the end of the arm. Therefore, if it isn't bigger than the end part of the arm it can easily be stuck on. However, you can take measurements if you want which will make the CAD more realistic to your components.

Step 1.3.3: Set aside the electromagnet for use in later steps.

Step 1.4: Adjust CAD

Putting all your measurements of your motors and gears into the CAD can sometimes cause errors in the resulting CAD assembly. For example, if the stepper motor is too large, the base dimensions may have to be adjusted to compensate it. If this is the case, you should use your own better judgement to adjust the errors in the CAD. For example in this case the base assembly parts would need to be adjusted.

Step 2: ABS frame assembly

Step 2.1: Printing templates

To begin with, open the OnShape document once more and select either the stepper or DC motor version you wish to use. The variable studio allows for the adjustment of many variables, although we recommend you leave any variables unchanged if you are unsure what the effect of changing them might be. For instance, lengthening the linkages might cause the motors to be underpowered and unable to drive the linkage.

There are two different sheet thickness options if you can’t find enough sheets in one size. One thickness for the linkage and one thickness for the base. Although all the panels we salvaged were 3mm thick.

Now access the templates tab on the CAD and print all the templates on your working printer.

Step 2.2: Cutting shapes

Cut these templates out roughly using any sharp tool such as scissors or a Stanley knife, it is best to cut them about 5mm from the line. Now mount them to the ABS sheets you have salvaged, preferably using a glue stick rather than anything stronger as these will need to be removed after cutting. Ensure that any non-flat surfaces are those that do not interfere with other flat surfaces. If you are unsure, refer to the example image to see which surfaces should be flat.

Now cut out all templates with a wood or metal saw, close but not to the template lines. Then mount the roughly cut pieces into a vice if you have access to one and use a file and sandpaper to shape the parts to their final dimensions (to the lines). If you don't have access to a vice, then by hand works as well.

Before removing all the paper templates from the parts, grab a centre punch and punch the centre of all the holes with a hammer on the parts which are clearly marked. If you do not have access to a centre punch this can be done with a simple wood screw as the soft ABS is very compliant. Now the templates can be removed which is best done by wetting the parts with warm water if a glue stick was used or by using sandpaper if slightly stronger glue was used.

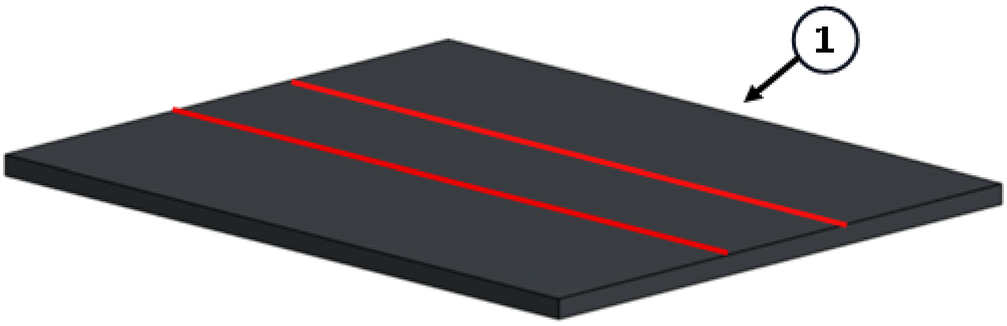
Finally, with a power or hand drill into the base parts and linkage arms at the locations you marked out previously. The size of these holes however will depend on the size of the screws that you have salvaged from your printer. If your printer contains some nuts and bolts like ours you can select linkage joints for those and drill the holes to the diameter of the bolts. However, if only screws are to be used one must ensure that the pivot that receives the screw thread is small enough for the threads to bite so that they can have some movement.

Step 2.3: Base Assembly

Look at how the pieces for the base fit together on the Onshape CAD model before attempting to glue them together. The red parts in all the visual images that help with this assembly process are the new parts that have been introduced in that step. Any transparent parts are there to help visualise the combining of parts better. Please read the whole step before attempting to ensure you understand. Some of the parts may be put together differently in later steps but this shouldn't affect the functionality of the robot.

Step 2.3.1

Use a ruler and a set square to mark out the lines with a Stanley knife of where the pieces will be placed on the bottom of the base (1). The red lines indicate where the middle pieces will go and they should be 30mm apart.

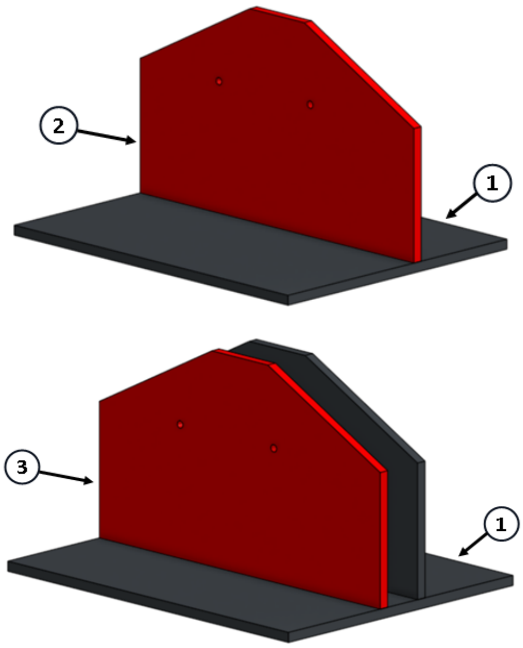


Step 2.3.2

Firstly, rough up the areas that are going to be glued on parts 2 and 3 using sandpaper or a file to help the adhesion process. Also, do this on the inside of the red lines on part 1.

Now mix up some epoxy glue on a scrap piece of plastic or paper. Apply it to part 2 and stick it to part 1 with the edge on the inside of the red line, use a set square or another known 90-degree angle tool to prop it up against to ensure that it is at the correct angle. Apply pressure till it has cured. Repeat this for part 3, but on the inside of the other red line.

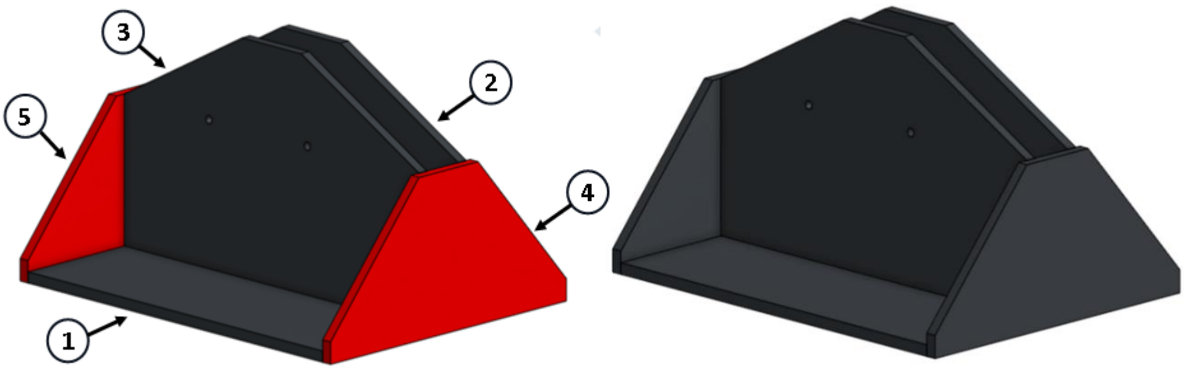
Hint: An easy technique to ensure the sides are evenly spaced when glueing part 3 is to use paper as a spacer between the two. The use of clamps or rubber bands can help clamp the pieces better in place as well.



Step 2.3.3

Firstly, make sure that the previous batch of epoxy hasn't dried up, if so mix some more. Similar to the middle parts, rough up the areas that are going to be glued on parts 4 and 5 as well as the edges of parts 2 and 3. Apply epoxy to the areas on part 4 and attach them to the side of the base. Then repeat this for part 5 immediately after, make sure to apply pressure to both and use a rubber band or clamp to hold them both in place.

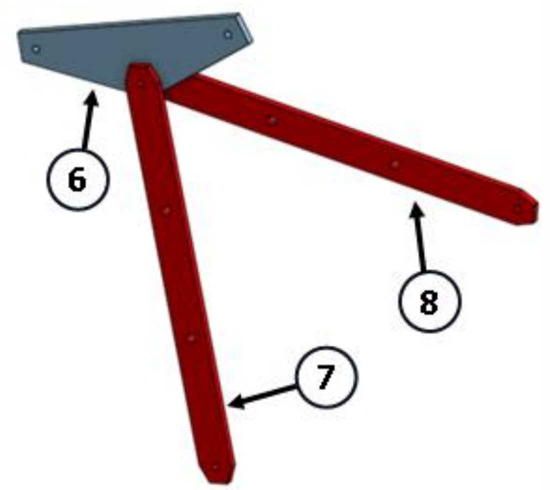
After the epoxy has cured, add a filet of epoxy to all the joint lines to further strengthen the joints. The base is now complete for now.

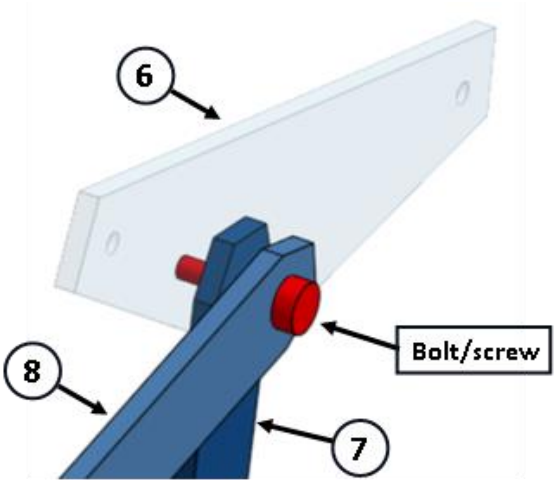


Step 2.4: Linkage Assembly 1

Step 2.4.1

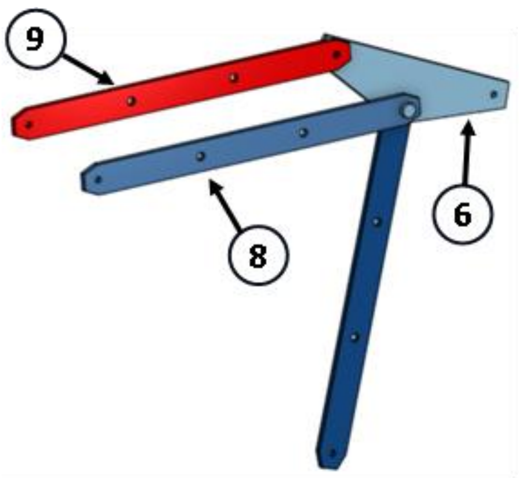
For the first half of the linkage assembly start off with part 6. Align the end hole of part 7 over one side of the middle hole of part 6 and the end hole of part 8 over the other side. Now put a bolt or screw through these holes of parts 6,7 and 8 to hold them together starting from the side of the part 8 hole.

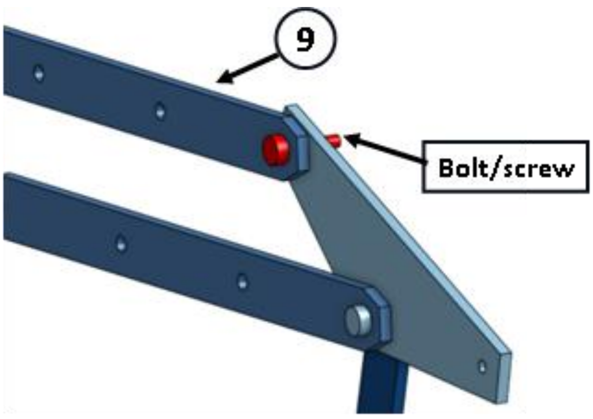




Step 2.4.2

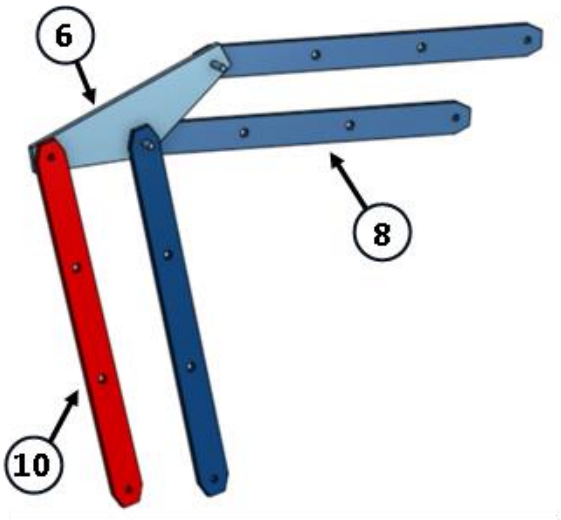
Align one of the end holes of part 9 over the left hole of part 6, part 9 should be above and parallel to part 8. Make sure the orientation and holes are correct by referring to the visuals. Then put a bolt or screw through these holes of part 6 and 9, going through part 9 first.

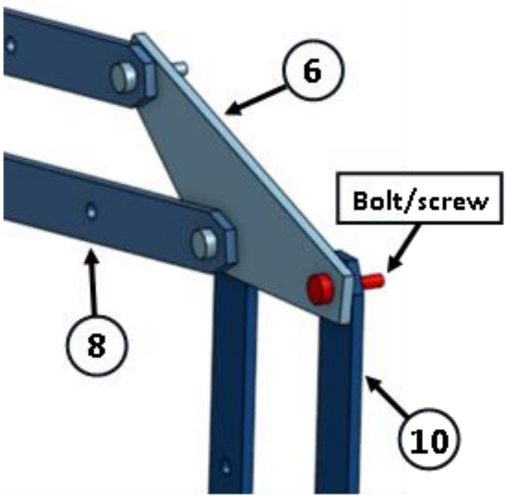




Step 2.4.3

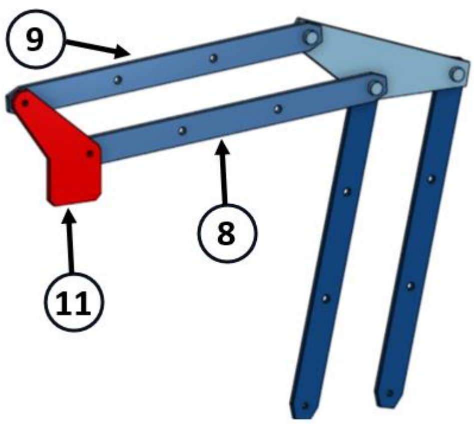
Flip the assembly and align one of the end holes of part 10 over the last empty hole of part 6, part 10 should be above and parallel to part 7. Make sure the orientation and holes are correct by referring to the visuals. Then put a bolt or screw through the holes of parts 6 and 10, going through part 6 first.

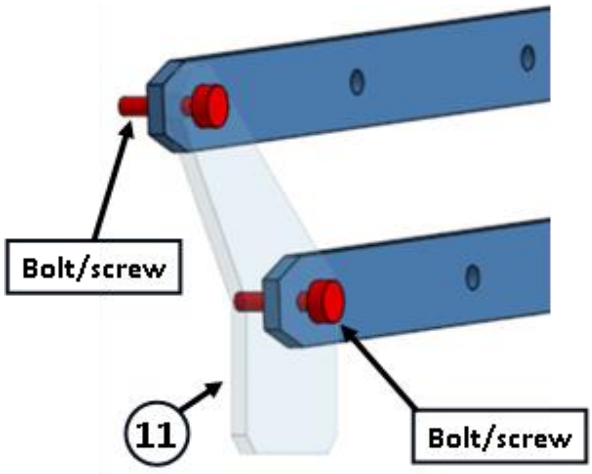




Step 2.4.

Now add the magnet holder part. Align the other end hole of part 9 over the top hole of part 11 while aligning the other end hole of part 8 to the hole halfway up part 11. Make sure the orientation and holes are correct by referring to the visuals. Then put a bolt or screw through the holes of parts 11 and 9, going through part 11 first. Do the same for the holes of part 11 and part 8, going through part 11 first again.

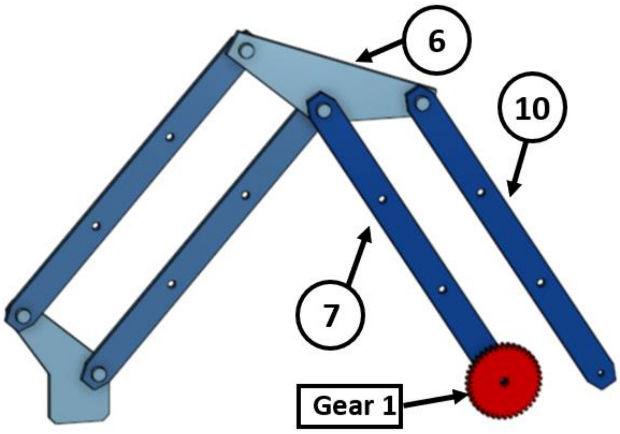




Step 2.4.5

Lastly add gear 1 to the empty bottom hole of part 7. This can be done by using epoxy glue and making sure that the holes of part 7 and gear 1 are aligned. Ensure not to get any glue in the gear teeth as this will affect the robot's functionality later. You have now completed linkage assembly 1.

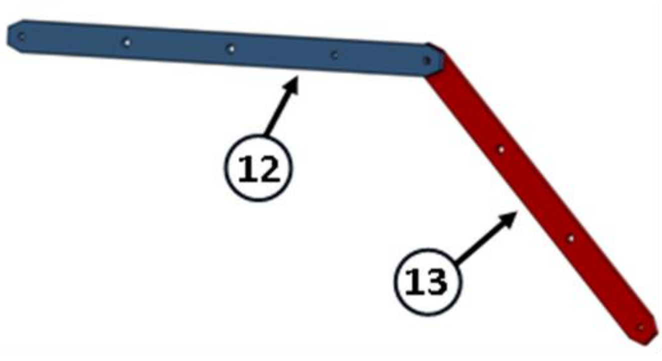
Hint: Sometimes putting an extra screw or bolt through the body of gear 1 and into part 7 can extra strengthen gear 1 so it doesn't move.

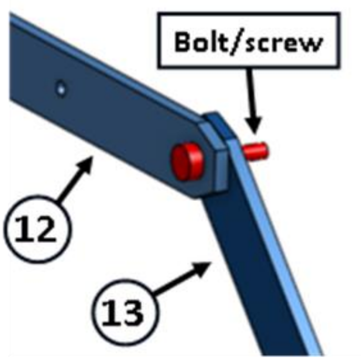


Step 2.5: Linkage Assembly 2

Step 2.5.1

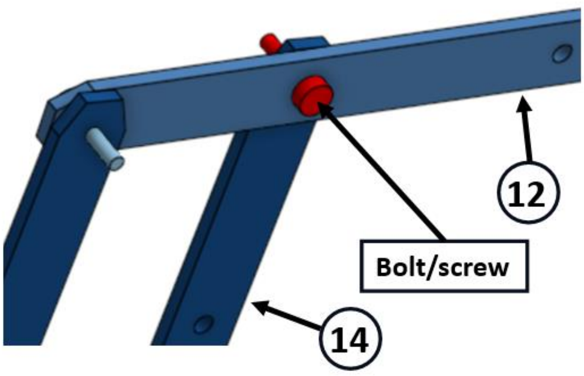
For the second half of the linkage assembly start off with part 12. Align one of the end holes of part 12 over one of the end holes of part 13 and put a bolt or screw through these holes going through part 12 first.

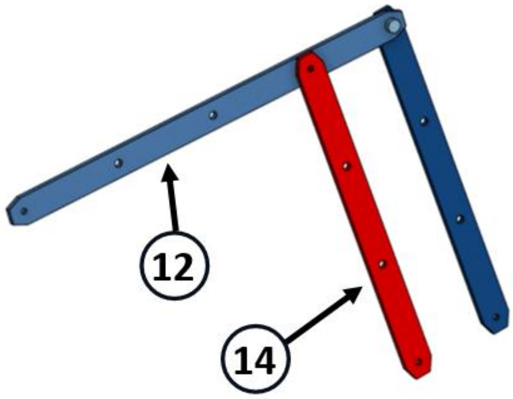




Step 2.5.2

Now align one of the end holes of part 14 over the next hole from part 13 but on the other side. Then put a bolt or screw through these holes going through part 14 first. You have now completed part 2 of the linkage assembly.

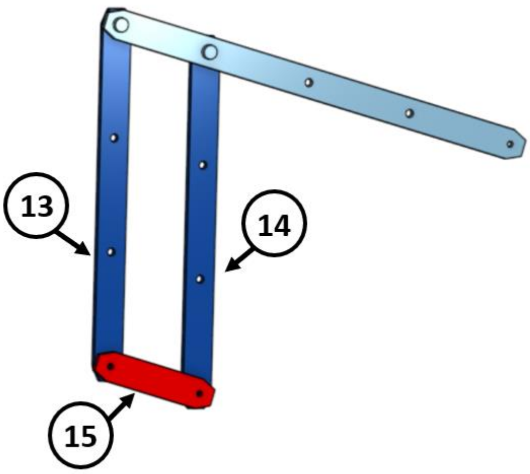


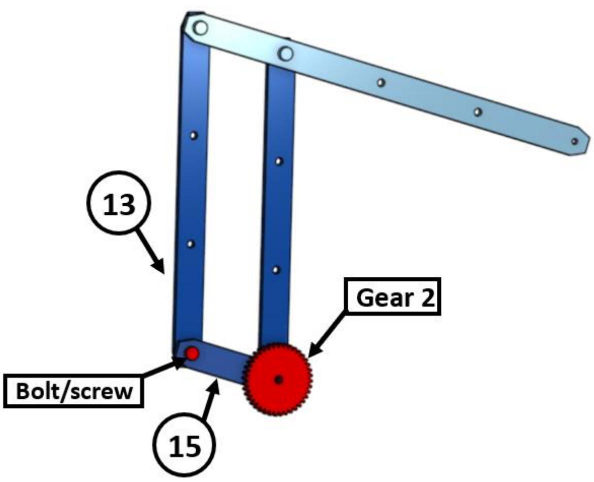


Step 2.5.3

Align one hole of part 15 over the empty end hole of part 13, and the other hole of part 15 over the empty end hole of part 14. Then put a screw or bolt through the aligned holes of part 13 and 15 going through part 15 first. Lastly, attach gear 1 to the empty hole of part 15. This can be done by using epoxy glue and making sure that the holes of part 15 and gear 2 are aligned. Ensure not to get any glue in the gear teeth as this will affect the robot's functionality later. You have now completed linkage assembly 2.

Hint: Sometimes putting an extra screw or bolt through the body of gear 2 and into part 15 can extra strengthen gear 2 so it doesn't move.

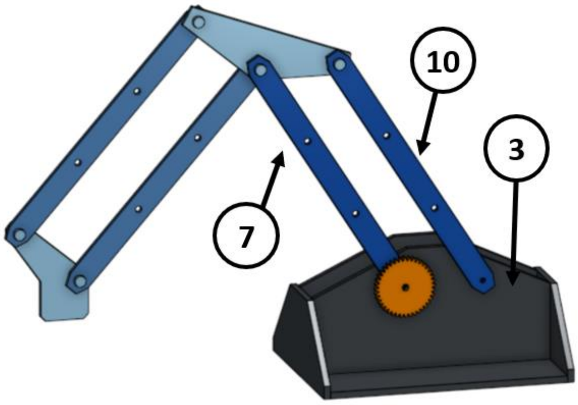


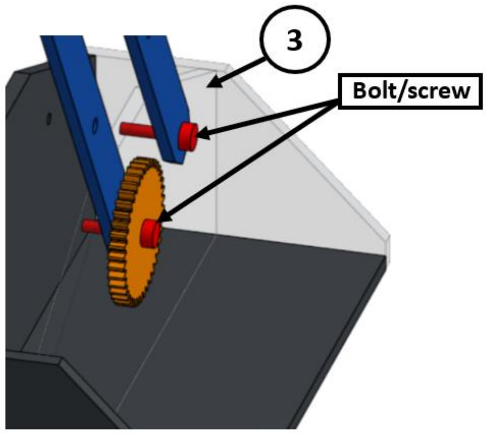


Step 2.6: Attaching linkage to base

Step 2.6.1

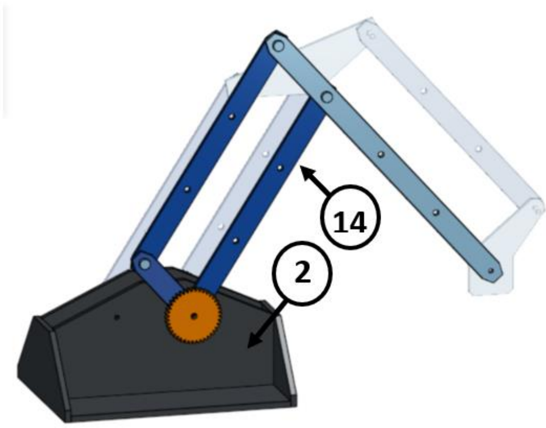
Take linkage assembly 2 which is already made and align the empty end hole of part 7 over the left hole of part 3. Then put a screw or bolt through these two aligned holes going through the gear and part 7 first. Now align part 10 over the right hole of part 3 and put a screw or bolt through these two aligned holes.

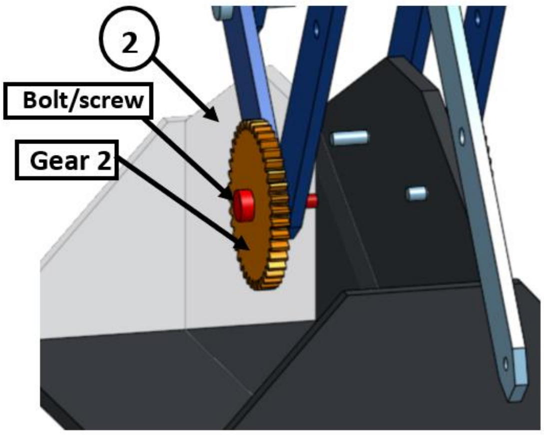




Step 2.6.2

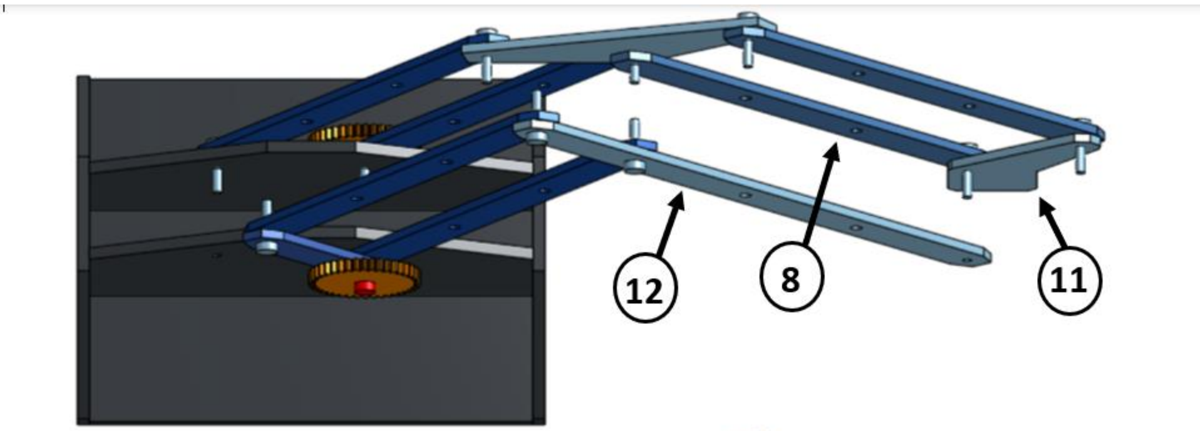
Turn the robot 180 degrees. Take linkage assembly 1 which is already made and align the empty end hole of part 14 over the right hole of part 2. Then put a screw or bolt through these two aligned holes going through gear 2 and part 14 first.

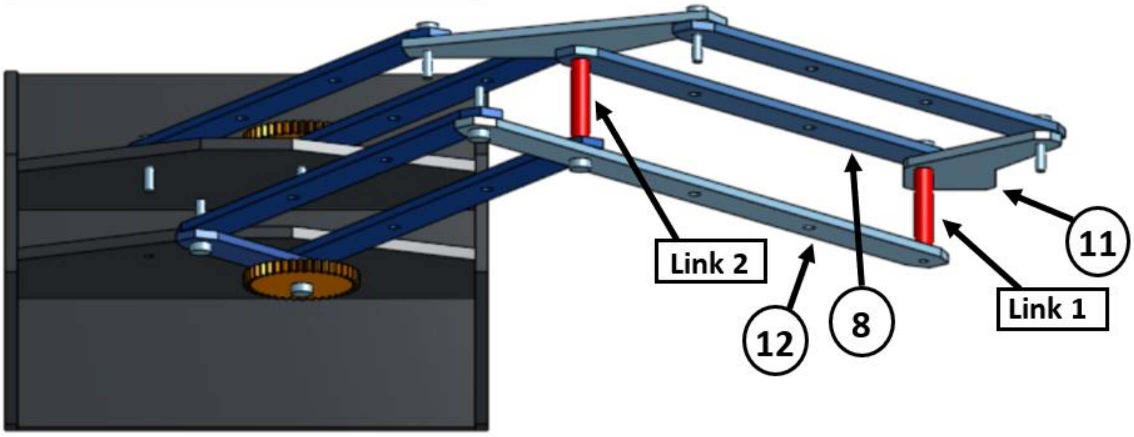




Step 2.6.3.

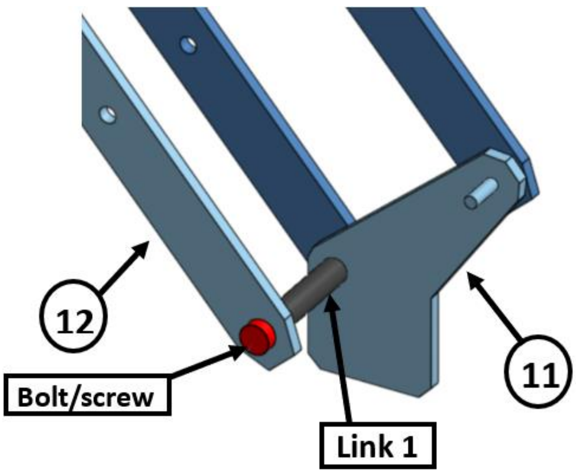
These arms now need to connect. This is done with link connectors which are thin plastic tubes with a hole through the middle found in the printer. The dimensions of these are found within the CAD. Starting with link 2, carefully screw it onto the screw or bolt going through part 8 first and then into the screw going through part 12. Then for link 1, screw it into the screw or bolt going through part 11.





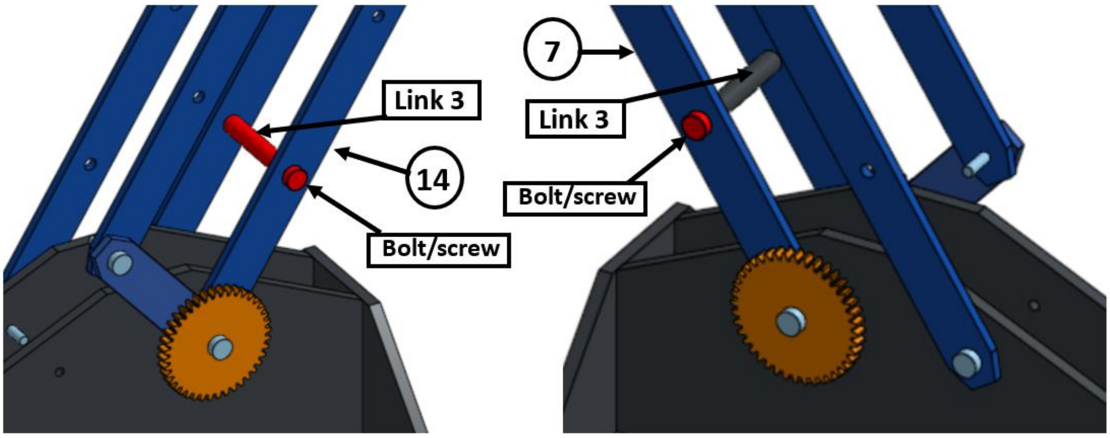
Step 2.6.4

Now grab a screw or bolt and put it through the empty end hole of part 12 into link 1.



Step 2.6.5

The final link connector goes between parts 14 and 7. Put a screw or bolt through the hole above the gear going through from the side of the gear on part 14 and then screw link 3 onto the end of this screw. Flip the robot 180 degrees and put a bolt or screw through the hole above the gear on part 7 into link 3. The ABS frame assembly is now complete.



Step 3: Motor attachment

Now it is time to attach all the electrical components to this ABS frame assembly. Therefore the DC case motors which you have emended and the electromagnet from Step 1 will be needed in this step.

Step 3.1: DC Case Motors

Detailed explanation:

Read this to better understand how the motors are attached as step 3.1.1 and 3.1.2 only tells the location of the placement and orientation.

This is one of the more precise operations required in the assembly of the robot as the gears must mesh properly for the robot to function as intended with minimal play and tolerance. Before beginning this step, ensure that the linkage-mounted gears are solidly fixed and are not able to translate (move in x or y) on their axis. If so, check whether all screws or bolts are tight and if the gear is solidly fixed to its shaft. Tape can be used on the shaft to ensure a tight fit if not.

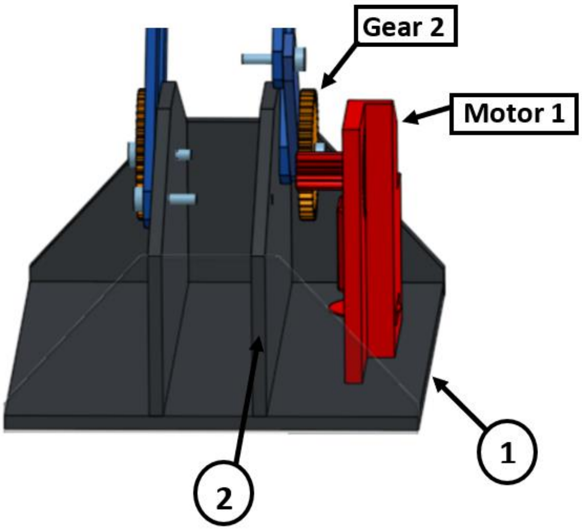
To mount the DC motors, it is recommended to mount the motor with the lowest gear ratio to the linkage controlling the shoulder as this is the joint requiring the greatest torque. To mount the motors, find an orientation using the flat sides of the case that allows the pinion and drive gear to mesh well whilst maintaining a solid orientation on the motor.

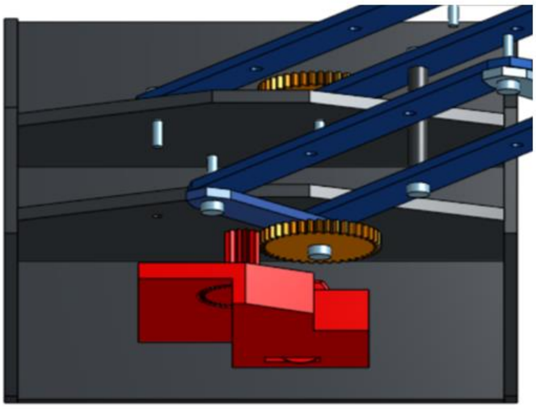
Next, apply a small amount of hot glue (or any other non-permanent fast-drying glue) to the surface of the motor being attached to the base. Make sure the pinion and drive gear are well meshed as the glue is drying. Now one can remove the glue residue and refasten the motors in place.

Hint: After the glue has dried, drill two holes through the bottom of the base into the motor unit and screw in two screws ensuring they are the correct length so as not to interfere with the encoder or motor circuit board.

Step 3.1.1

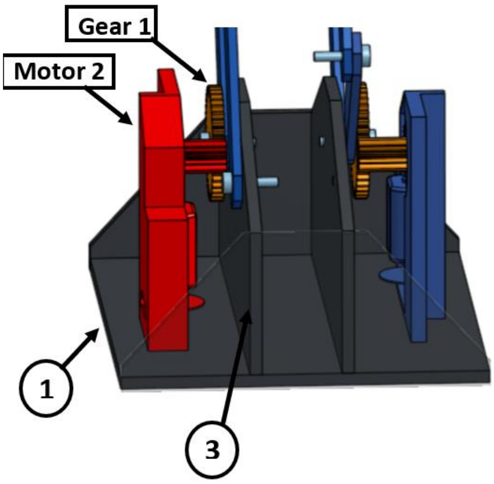
Take one of the case motors and attach it to the base of the robot, part 1 on the side of part 2. This can be done by using glue. Make sure when doing this that the gear from motor 1 meshes and aligns with gear 2. Look at the visuals to fully understand the orientation.

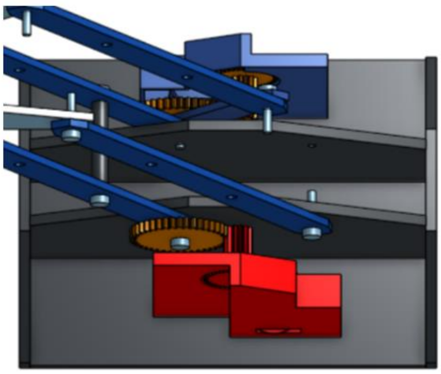




Step 3.1.2

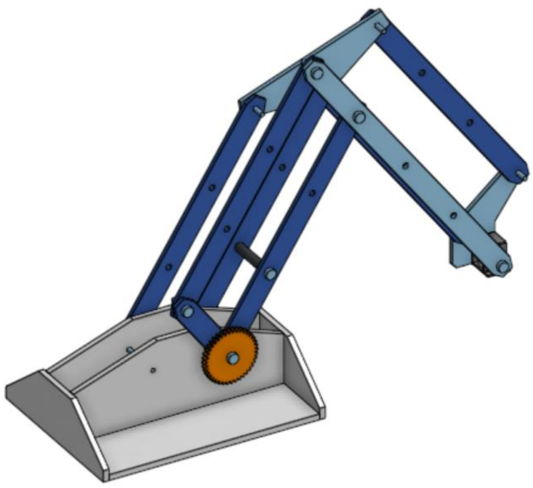
Take the other case motor and attach it to the base of the robot, part 1 on the side of part 3. This can be done by using glue again. Make sure when doing this that the gear from motor 2 meshes and aligns with gear 1. Look at the visuals to fully understand the orientation.





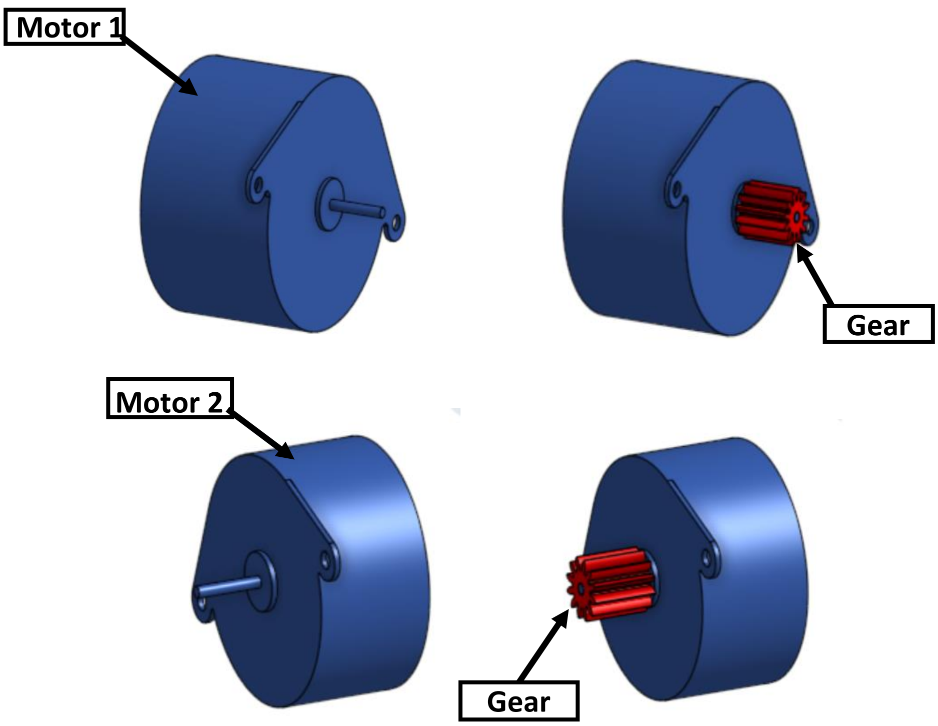
Step 3.2: Stepper motors

If you want to mount stepper motors there are several options available. One can simply epoxy in place the stepper motor, ensuring that the gears are well meshed. Or otherwise, to preserve the ability to remove the steppers one can use the integrated mounting bracket in the stepper motor to screw these into the sides, using appropriate spacers to maintain gear meshing. The visual below displays the stepper version ABS frame assembly which is made in the exact same way as the DC version ABS frame assembly. The only slight difference is the dimensions of the base of the assembly which are also found on the CAD.



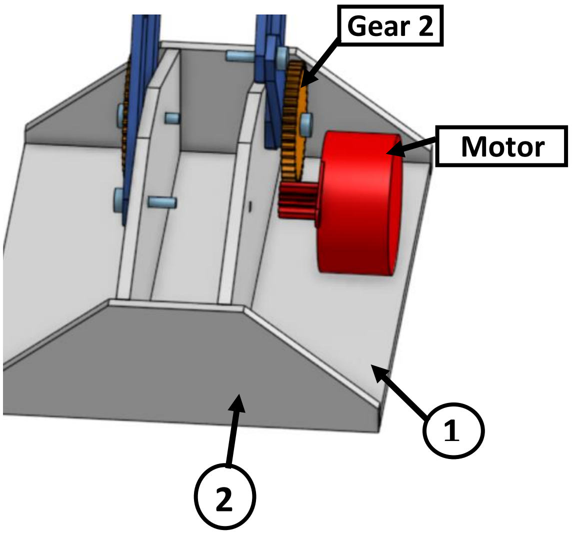
Step 3.2.1

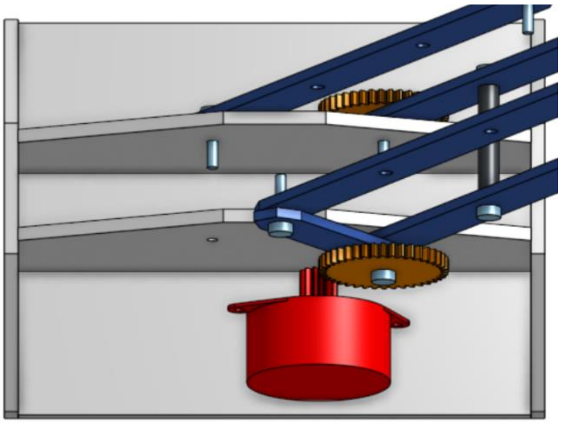
To begin with take motor 1 and attach the small gear to it, this should be the same gear ratio as the DC version of the robot. Do the same for Motor 2. Glue sometimes makes it stay on better if it doesn't fit neatly on. For the motor that we found, one was bigger than the other, in this case motor 1 is bigger than motor 2.



Step 3.3.2

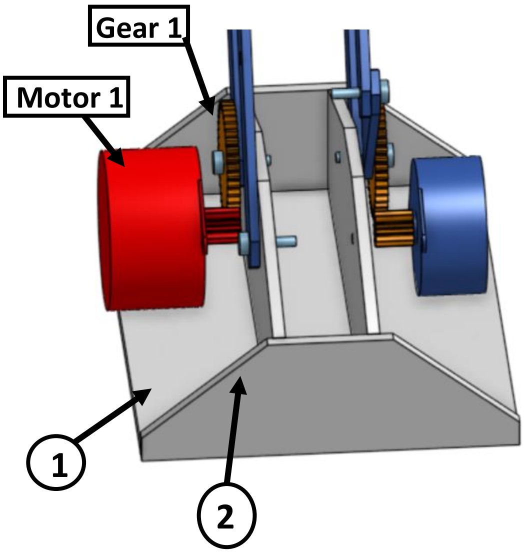
Take one of the stepper motors and attach it to the base of the robot, part 1 on the side of part 2. This can be done by using glue. Make sure when doing this that the gear from motor 2 meshes and aligns with gear 2. Look at the visuals to fully understand the orientation.

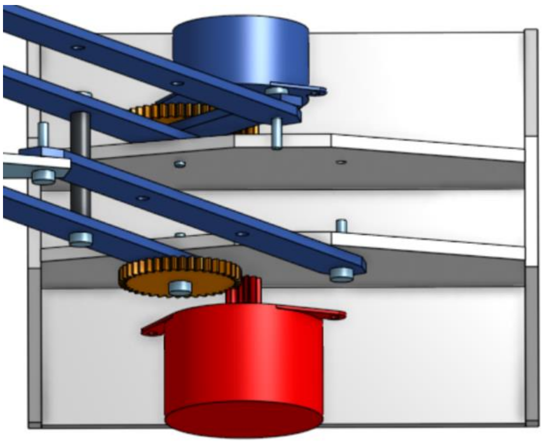




Step 3.3.3

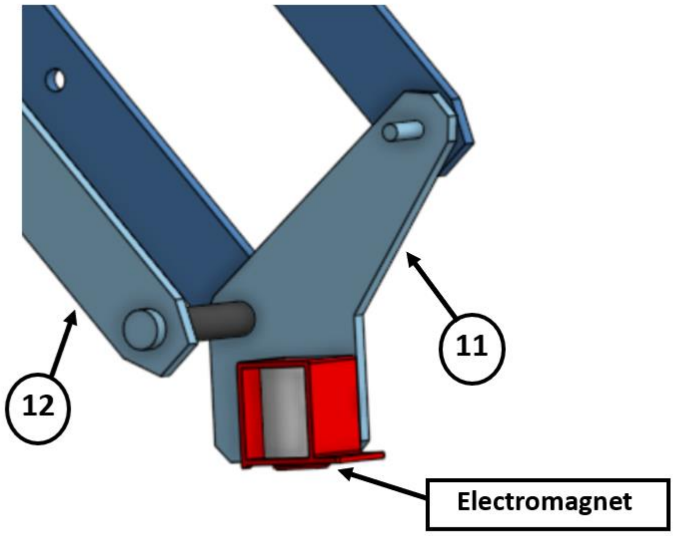
Take the other Stepper motor and attach it to the base of the robot, part 1 on the side of part 3. This can be done by using glue again. Make sure when doing this that the gear from motor 1 meshes and aligns with gear 1. Look at the visuals to fully understand the orientation.



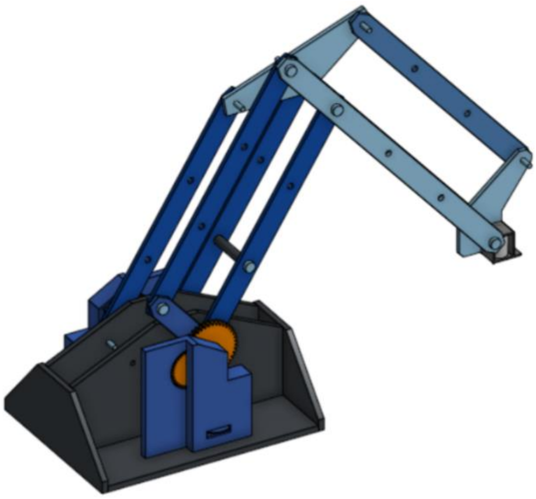


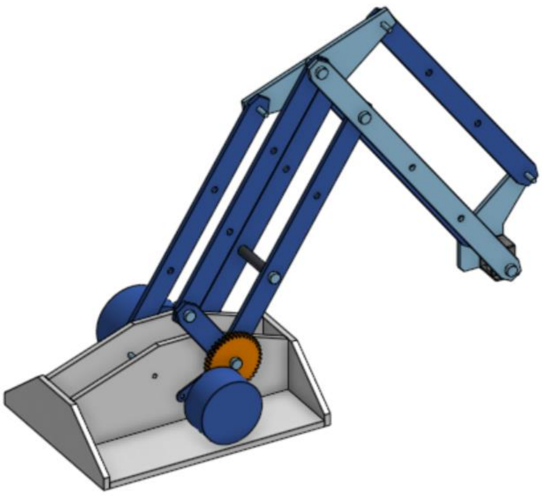
Step 3.3: Electromagnet

Finally using epoxy glue attach the electromagnet to the end of the linkage on the bottom of part 1, with the direction of the magnet pointing downwards



Step 4: The manufacturing part of the robot kit is now completed.



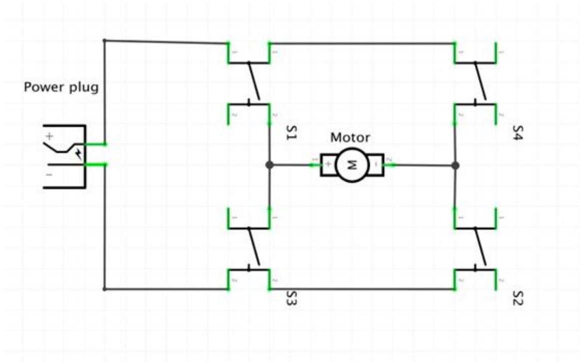


5.4 Creating the Analogue Control Circuit

5.4.1 Controlling Brushed DC Motors

Getting a brushed DC motor to move is a straightforward task. You connect the positive side of a power source to one wire (or pin) and the negative side to the other, and your motor will start spinning. If you want the motor to move the opposite way, simply switching the wires will make it spin the other way around. However, if you need to reverse the motor's direction, swapping the wires may not suit applications like robotic arms. Instead, you can use other components from your electronic waste to make control more manageable.5.4.2 Switches for Control

One effective way to control a brushed DC motor is by using switches. There are various types of switches, with the simplest being on-off switches (also known as Single Pole Single Throw switches). These on-off switches are commonly found in electronic devices, making them a practical choice for motor control. We'll illustrate how to use them to control a motor's rotation in both directions by creating something known as an H-bridge circuit.



5.4.3 Creating an H-Bridge Circuit Controller

The H-bridge circuit is named after its configuration, which resembles the letter "H." It uses four switches arranged to control the motor's direction. When switches 1 and 2 are closed, the circuit is complete, and the motor spins in one direction. Conversely, if switches 3 and 4 are closed, the motor rotates in the opposite direction, providing full control over its movement.

Step 1: Connect one output of your power supply to one side of switches 1 and 3.

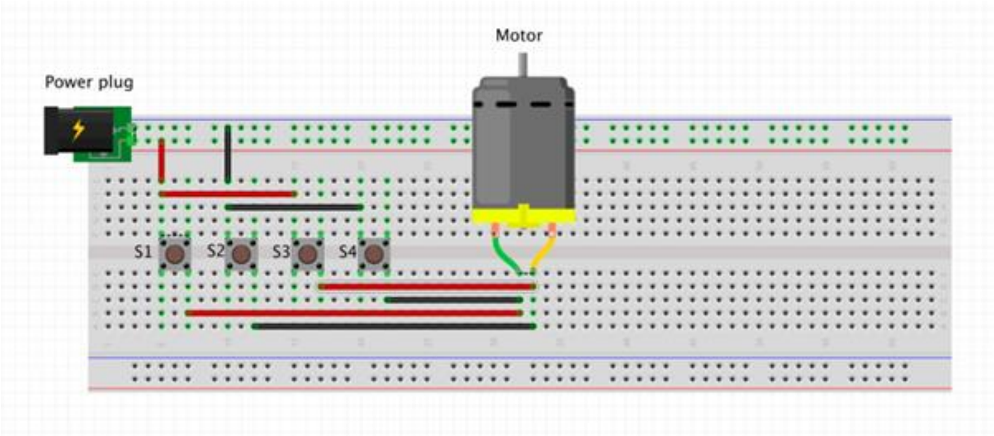
Step 2: Connect the other output of your power supply to one side of switches 2 and 4. Whether the output is negative or positive does not matter for the working of the motors, it just influences which direction it will spin when either of the two switches are pressed.

Step 3: Connect the output of switches 1 and 4 to one side of the motor.

Step 4: Connect the output of switches 2 and 3 to the other side of the motor.

Step 5: You're done! To move the motor one way, press buttons 1 and 2 together. To move it the opposite way, press buttons 3 and 4!

You can follow the diagram below to help you with the wiring.



Avoiding Short Circuits

It's essential to avoid closing switches 2 and 3 or more than 2 together. Doing so creates a short circuit, where there is no resistance in the circuit, causing excessive current flow. This can lead to overheating of the wires and, if a battery is the power source, potential leakage or even an explosion. To protect the circuit, the battery, and yourself, refrain from creating these short circuits.

Running Multiple Motors and Electromagnet

If you need to control more than one motor, you can create additional H-bridge circuits or wire the motors to separate switches for turning them on and off. However, this method does not allow multiple motors to move in different directions simultaneously.

Step 1: Instead of connecting the output of the switches to the motor directly, connect the output of switches 1 and 3 to one side of two additional switches (named MC 1 and MC 2 in the manual).

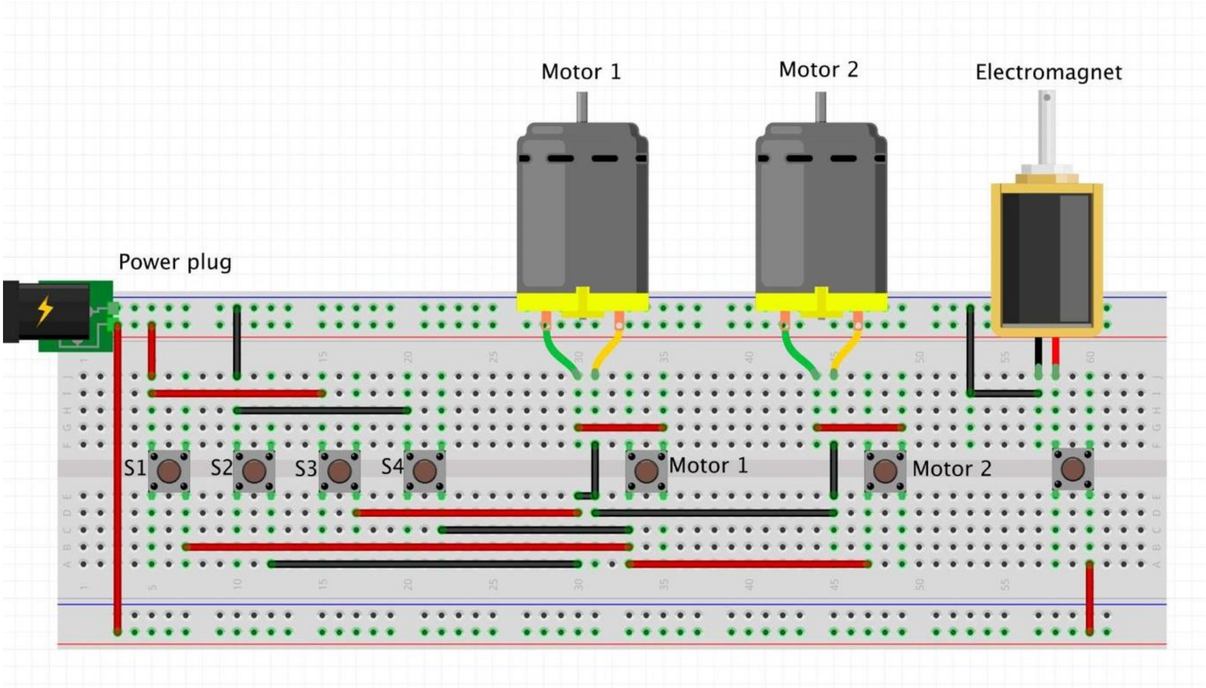
Step 2: Connect the output of switches MC 1 and MC 2 to one leg of Motor 1 and Motor 2. The leg which you choose does not matter.

Step 3: Connect the output of switches 2 and 4 to the other leg of Motors 1 and 2. You now have control over two motors at the same time with just two more switches! If you wish to connect more motors, simply add more switches along with MC 1 and 2 and wire the outputs of switches 2 and 4 the same way to the new motors.

Step 4: To connect the electromagnet, wire one of the outputs from the power supply to one side of a switch and the other side to one of the wires on the electromagnet.

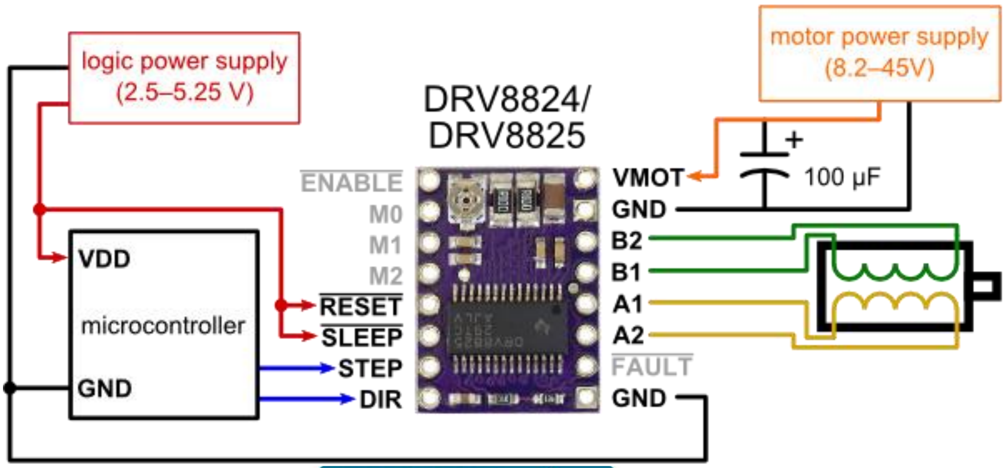
Step 5: Connect the other side of the switch to the other wire of the electromagnet! Now you have control over multiple motors and an electromagnet using only switches and wires!

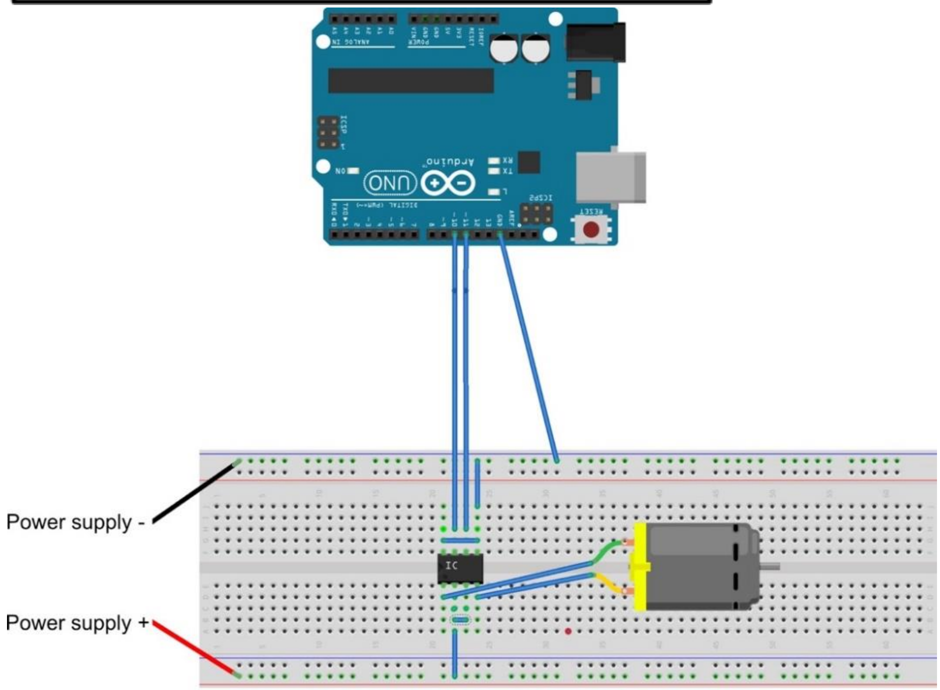
You can follow the diagram below to help you with the wiring.



6. Arduino control of the robot

Another way of controlling the robot is using a microcontroller board. We suggest an Arduino Uno for this. Using a microcontroller allows you to pre-program sequences of movements which the robot then executes. There are two ways of interfacing the Arduino with the robot, depending on whether you are using DC or stepper motors. For both, you need an additional driver chip. We recommend the DRV8824 driver for steppers and the L9110H driver for DC motors. There are two circuit diagrams below that show how to connect a stepper or a DC motor to the Arduino. You can use any control pins on the Arduino for the STEP/DIR pins in the stepper case or forward/backward pins in the Arduino case, in the DC diagrams pins 10 and 11 are used. There is a stepper motor wiring diagram below.





Example DC motor wiring diagram - here using Arduino control pins 10 and 11. This circuit does not show the encoder wiring.

However, in the DC motor case, you also need to wire the encoder pins to the Arduino so the Arduino can track the motion of the motors and arms. An example of the complete DC wiring diagram follows:

6.1 Example Operations

We have prepared multiple examples of functions that can be used with a microcontroller to operate the robot arm, but creativity is your limit!

Arduino uses its own version of the C++ programming language but do not worry, you don't need to know much programming at all to get the robot running and understand how it works.

There are three main concepts that are needed in order to understand the code- classes, objects and functions. A class is like a container for the object, one that holds all the common properties of the different objects you might have. As in illustration, think of a class of cars. An example of a car might be a blue Toyota sedan, that is our object, but there are many different cars, so in our class, we can store the colour, brand and type of car! A car class definition might look as follows: car(str colour, str brand, str type) and a car object

Functions are quite self-explanatory, they act on the object and perform a set of instructions, or one might say a "function". Following our car example, we might give the car object a move function which would cause it to move to a given position. These are called by using a "." after the object, for example car.move().

6.2. DC Motor Class

```
myMotor(int powerPin1, powerPin2, int enc1pin, int enc2pin, float calibAngle)
```

This is the code responsible for creating our DC motor object. Depending on the number of motors, you will need to create an appropriate number of DC motor objects, each one corresponding to a real motor.

powerPin1 and powerPin2 are the pins responsible for controlling the motor driver. For more information about the pinout look at the relevant schematic, in the case of the L9110H, see https://www.adafruit.com/product/4489.

Enc1pin and enc2pin are the wires responsible for the encoders, in the specific case, the blue and yellow wires.

calibAngle is the angle that needs to be set for calibrating the robot. A value that is too small might reduce the accuracy while a value too big might lead to the robot parts colliding. We used 22.5 degrees, which is a good place to start with your robot.

```
myMotor.zero()
```

This function sets the new zero position for the motor. After running it, the encoder will output 0 at this position. The user needs to manually send commands through the Serial Monitor to set the zero position. The commands are of the format '+1234' or '-1234', with 1234 here being the number of steps to turn the motor, and '+' or '-' in the front to determine rotation position. If you want to set less than 1000 steps, you need to keep the 0 in the front like this: '+0123', '+0012', '+0001'. Finally when you are done, send 'allOK' to set the new position.

```
myMotor.calibrate()
```

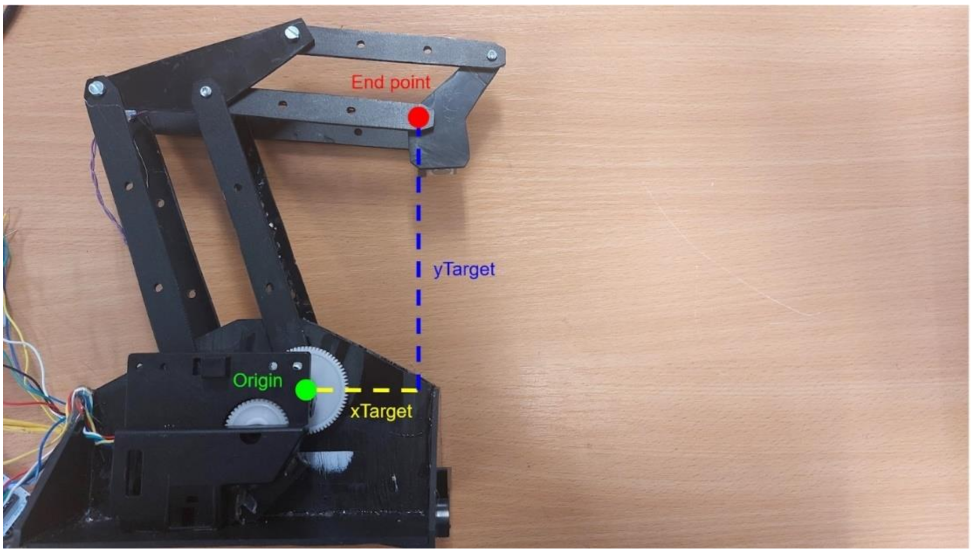
This function is responsible for determining how many steps correspond to a given angle. With this command, you move the motor to a predetermined angle in order for the Arduino to measure how many steps it takes to reach this angle, and as a result control the motor to precise angles.

The angle to which you need to rotate the motor is set by the parameter calibAngle. To move the motor you need to enter commands of the same format as with the myMotor.zero() function ('+1234' or '+0123' or '-1234' etc.).

When you move the arm, measure the angle of the motor with a protractor, and once you have reached the desired angle, send 'allOK' through the Serial Monitor.

```
myMotor.moveToXY(motor m1, motor m2, float xTarget, float yTarget)
```

This function moves the end point of the robot to a given coordinate point relative to the root point of the robot (as shown on the image below). It takes multiple inputs, motor m1 and m2 being the two motors responsible for the lower arm and the upper arm respectively. xTarget and yTarget are the given position you want the robot to move to. IMPORTANT: xTarget and yTarget must be in cm.



```
myMotor.moveNSteps(int stepsToTurn, int dir, int speed)
```

This function moves the motor a given number of steps (stepsToTurn) in a direction specified by ‘dir’, with a speed given by ‘speed’.

For ‘dir’, you can use either 0 for clockwise or 1 for counter-clockwise motion, or you can use the macros CW and CCW instead. For ‘speed’, use a number between 0 and 255. This parameter defines the duty cycle of the analogue signal of the pin.

```
myMotor.setAngle(float angle, int speed)
```

This function moves the motor to a particular angle in degrees. Just as before, for ‘speed’, use a number between 0 and 255. This parameter defines the duty cycle of the analogue signal of the pin.

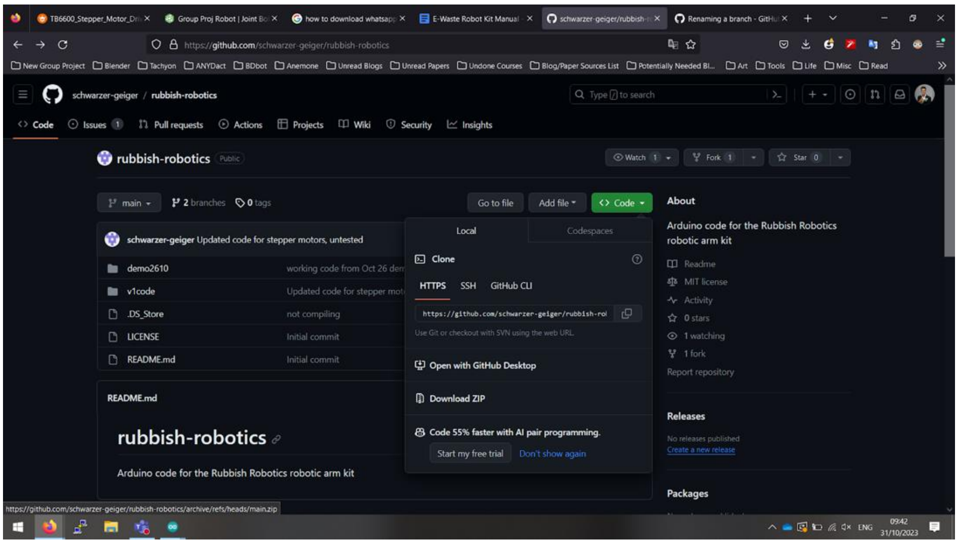
IMPORTANT: use only after calibration!

```
myMotor.manualMov()
```

This function allows for manual movement of the motor using the Serial Monitor in the same way as earlier, with myMotor.zero() and myMotor.calibrate().

The user needs to manually send commands through the Serial Monitor to set the zero position. The commands are of the format ‘+1234’ or ‘-1234’, with 1234 here being the number of steps to turn the motor, and ‘+’ or ‘-’ in the front to determine rotation position. If you want to set less than 1000 steps, you need to keep the 0 in the front like this: ‘+0123’, ‘+0012’, ‘+0001’. Finally when you are done, send ‘allOK’ to set the new position.

Our most up-to-date code is available at https://github.com/schwarzer-geiger/rubbish-robotics. The ‘main’ branch contains code that has been confirmed to work, while ‘experimental’ is for code with new but potentially unstable features.



Running the code on your robot

IMPORTANT: Before you start, make sure to set L1 and L2 in cm to match the L1 and L2 from the OnShape file. Make sure the units are correct! For example, if L1 is 170mm in Onshape, set L1 to 17 in the code.

We include a few example programs meant to test the motion of the robot. They can also be modified as needed.

7. Conclusions

We hope that this manual has helped you repurpose e-waste into something better, teaching you learn new things along the way and inspiring you to take a better look at the materials around you.

7.1 Further work

The robotic arm is just one example of what can be done with the various components that can be found in e-waste. The possibilities are endless:

- Adding additional degrees of freedom to the arm;
- Making a handheld controller for both the analogue robot and a digitally controlled one;
- Making racing cars or boats using the motors;
- Things that we cannot even imagine.

  Your creativity is the limit and once again we would love to see pictures of the robots you have built using our kit and anything else that we inspired you to do! Please reach out to notrubbishrobotics@gmail.com.