UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Industrial Engineering

Design and implementation of a remote controlled excavator using a microcontroller and 3D printing

End of Degree Project

Bachelor's Degree in Industrial Engineering

AUTHOR: Lomm Hedvold, Hampus

Tutor: Baraza Calvo, Juan Carlos

Cotutor: Gracia Morán, Joaquín

ACADEMIC YEAR: 2021/2022

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALE

DESIGN AND IMPLEMENTATION OF A REMOTE CONTROLLED EXCAVATOR USING A MICROCONTROLLER

AUTHOR: Hampus Lomm Hedvold

SUPERVISOR: Juan Carlos Baraza Calvo

SUPERVISOR: Joaquín Gracia Morán

Academic year: 2021-22

SELECT
MODE
START
1
2
3
4

Abstract

This project seeks to design and build a radio-controlled excavator using 3D-printing and a microcontroller. This project aims to investigate how a radio-controlled excavator can be built using 3D-printing technology for creating the parts and a microcontroller for connecting all the electronics. The final product should resemble a real excavator both in design and the way it is controlled.

The project is divided into three parts: design, programming, and assembly.

The design is made in CAD using the software Autodesk Inventor. A complete assembly is created in CAD including all parts before they are 3D-printed. Programming of the microcontroller for controlling all motors and linear actuators with the wireless controller is done using Arduino IDE.

The final phase is the assembly where 3D-printed parts are mounted together with the electronic components to form the final product.

The result satisfies the scope of the project. The excavator has all the main functions like a real one, including driving, rotating, and moving the arm. It is controlled using two joysticks together with a couple of buttons. Few amounts of flaws can be found, including not being able to turn.

Improvements for a next version includes upgrading the motors for driving, using the same linear actuators in all three places, and creating a lock preventing the rotational gear to fall apart.

Keywords
Mechatronics, Microcontroller, Radio-controlled, Linear Actuator, Excavator

Acknowledgements

A special thanks to the supervisors of this project, Professor Juan Carlos Baraza Calvo and Professor Joaquín Gracia Morán, for providing ideas, support, and knowledge during the project.

Table of Contents

List of Figures

List of Tables

Nomenclature

3D	Three Dimensional
DC	Direct Current
I/O	Input/Output
PWM	Pulse Width Modulation
RC	Radio Controlled / Remote controlled
CAD	Computer Aided Design
PS2	PlayStation 2

1 Introduction

This chapter includes a shorter explanation of the project. It includes the background, purpose, scope, and method of the project.

1.1 Background

Radio-controlled vehicles can take the shape of several types of vehicles, for example, cars, helicopters, tanks, airplanes, and boats. Many of the radio-controlled vehicles on the market are produced using injection molding. Nowadays, 3D-printing is rising in popularity among manufacturers and private hobbyists. This raises the question if it is possible to build a radio-controlled vehicle using only 3D-printing as the manufacturing technique for the parts and integrate it with electronics to control it.

1.2 Purpose

The purpose of this thesis is to investigate if it is possible to build a radio-controlled excavator using only 3D-printed parts and connect these to electric components to control it with the help of a microcontroller. The excavator needs to have the basic functionalities such as a moving arm, a body that can rotate 360 degrees, and be able to drive. The movement of the excavator needs to be controlled in the same way as a real excavator.

1.3 Scope

The scope of this thesis is to design and build a radio-controlled excavator controlled by a microcontroller via a wireless controller. With the final product including mechanical and electrical components working together. The duration of the project is approximately 3 months and for the budget there are no limits as long as everything is reasonable. A limitation and challenge for this project is the long delivery times making it difficult to get specific parts needed and last-minute changes had to be made.

1.4 Method

In the initial state of the project research was conducted about what parts could be used, different microcontrollers and methods of wirelessly controlling the excavator in order to achieve the final product. With this information the process of designing the excavator began. All parts to be 3D-printed were designed in CAD using Autodesk Inventor. After this the code for the microcontroller was written. At last, all parts were assembled to achieve the final product.

2 Theory

This chapter explains all theoretical information about the various aspects encountered during the project. It includes explanations about the electrical components and software's used.

2.1 Microcontroller

The microcontroller is a compact integrated circuit used in embedded systems. A standard microcontroller includes the processor, memory, and input/output pins(I/O). It can be described as a smaller computer and can be found in for example, robots and vehicles among other devices [1]. In this project it is used to interpret the signals sent from the wireless controller and send the commands to the motors to achieve what the user intends.

2.1.1 Arduino Mega 2560

For this project, the Arduino Mega 2560 Rev3 is used. This microcontroller was chosen for the project because of the number of pins it has. The Arduino Mega has 54 digital input/output pins of which 15 support Pulse Width Modulation (PWM). It also includes 16 analog input pins. It is powered by a power source between 6-20 volts, however, 7-12 volts are recommended. It has an operating voltage of 5V which is the voltage supplied by every pin. For a full specification of the Arduino MEGA see *Appendix F. Arduino MEGA technical specification*. For this project, the PWM pins will be used to control all the motors. PWM stands for pulse width modulation and it is used for getting analog results with a digital signal. The digital signal creates a square wave which represents either on or off. The average voltage created is proportional to the time the wave is on/high. This is known as the duty cycle. By changing the duty cycle, a voltage can be simulated between the maximum, 5V in this case, and minimum 0V. Repeating the pattern of these waves results in a steady voltage signal [2][3]. This is particularly useful when controlling speeds of motors. See *Figure 1* for examples of how the PWM signal affects the voltage.

C

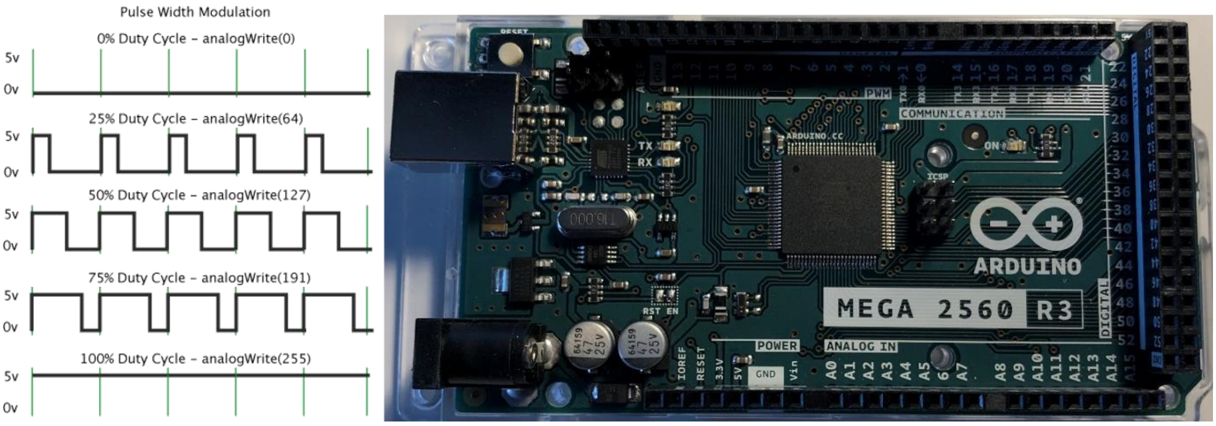


Figure 1: Example on voltage output depending on duty cycle and Arduino MEGA 2560 [3]

2.2 H-Bridge Motor Driver

A motor driver is a module used for controlling the speeds and direction of motors, in this case DC motors. It uses an H-Bridge to switch the direction of the current, and by doing this change the direction of the rotation. In this project the L298N motor driver will be used. This module can manage two DC-motors each with a voltage between 5-35 volts and currents up to two amperes. It is the chosen motor driver for this project because of the voltage it can manage and the high limit of current it can take.

The L298N is powered via 3-pin 3.5mm pitch screw terminals. These are: motor power supply (Vs), ground, and a logic power supply of 5V (Vss). There is an on-board 5V regulator that can be used with the help of a jumper. This means that if the module is powered with a maximum of 12V, and the jumper is in place, the logic power for the module comes from the motor power supply and the 5V input terminal can instead be used as an output for controlling the microcontroller [5].

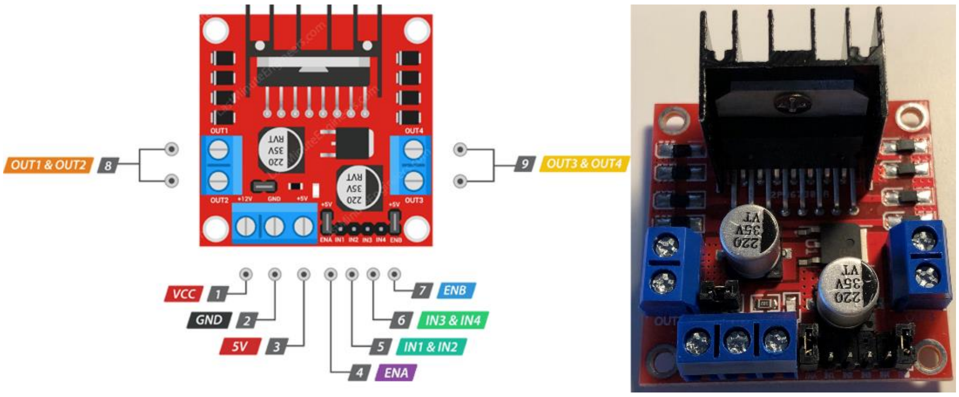


Figure 2: L298N motor driver [5]

For every motor there are two pins for controlling direction and one pin for controlling the speed. By applying a logic HIGH (5V) or a logic LOW (ground) to these direction pins the direction is changed. See *Table* 1 for examples.

Input1	Input2	Spinning Direction
Low (0)	Low (0)	Motor OFF
High (1)	Low (0)	Forward
Low (0)	High (1)	Backward
High (1)	High (1)	Motor OFF

Table 1: Different directions of the motor depending on the inputs [5]

The speed of the motor is controlled by sending a PWM signal from the Arduino to the speed pin, known as the enable pin. Using the equation below, the exact voltage output can be determined. The input value is the number of the PWM signal sent from the Arduino to the motor driver, and 11.1 is the voltage coming from the batteries.

$$Voltage = \frac{11.1}{1023} * Input\ Value \qquad (1)$$

With a motor driver between the motors and the Arduino, the voltage and current needed for the motors can be higher than the Arduino can manage. An external battery is connected supplying the motors directly.

2.3 DC-Motor

A direct current (DC) motor converts electrical energy of direct current to mechanical rotation. The motor consists of two main parts: the stator and the armature/rotor. The stator is stationary, and the armature rotates. When voltage is applied a magnetic field is generated in the stator. The rotor starts to rotate because of the attraction and repulsion of the magnets on the rotor. When voltage is increased the torque created also increases [6]. In this project a motor called TT-motor will be used for driving the excavator, and another DC-motor for the rotation. The TT-motor has integrated gears and two shafts coming out of it on either side.

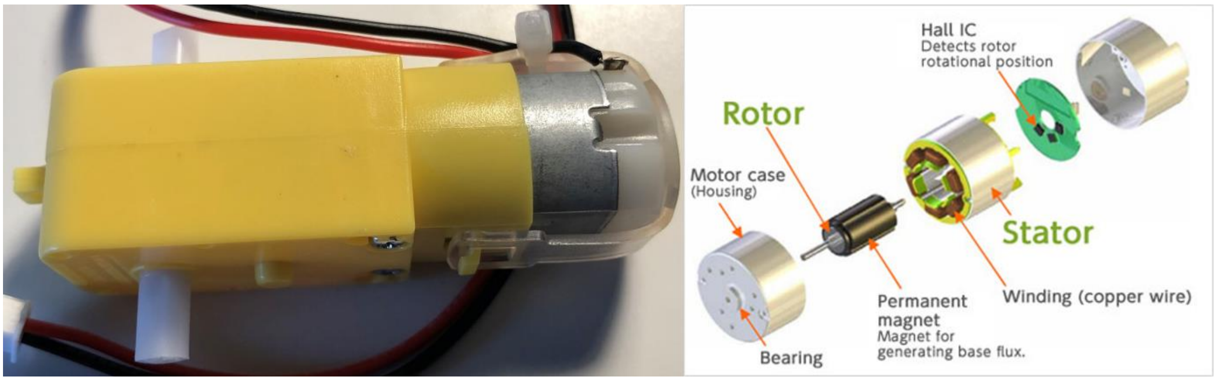


Figure 3: TT-motor and description of internal parts [15]

2.4 Linear Actuator

A linear actuator uses either a DC or AC motor and converts the rotary motion to a linear motion, which allows it to push and pull. The torque produced from the motor is transferred to turn a lead screw which in turn creates the linear motion. The direction of the motion is changed by changing direction of the current. One way creates a push motion and reversing the polarity creates a pull motion [7]. In this project micro linear actuators are used, which work in the same way as normal actuators but are even smaller and more compact. To control the direction and speed of the actuator, the L298N motor driver will be used.

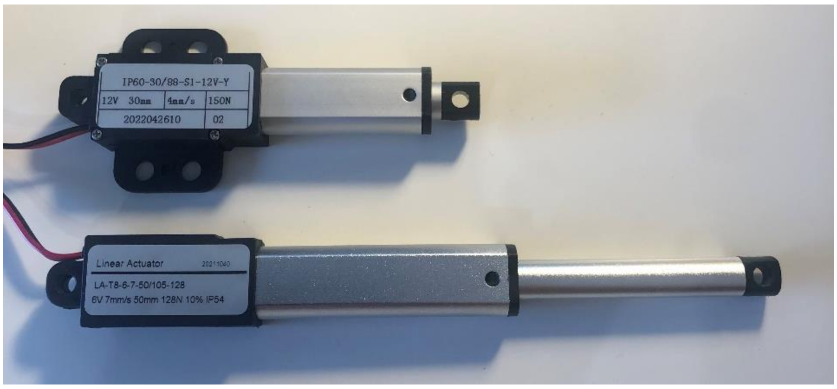


Figure 4: 30mm & 50mm linear actuator

2.5 Wireless PS2 Controller

The wireless PlayStation 2 controller is the main controller which the user will use to control the excavator. It communicates using 2.4GHz frequency and consists of the controller and a receiver. The receiver connects to the Arduino and is powered with 3.3 volts, and the controller is powered by normal AAA batteries. The controller consists of two analog joysticks, 12 analog pressure sensitive buttons and five digital buttons. Two motors of varied sizes are embedded in the handle for vibration. The receiver has nine pins of which six are needed. Two of these are used for power and the rest for communication [8].

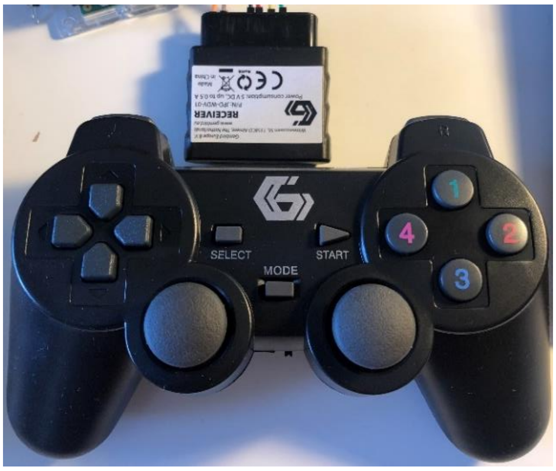


Figure 5: PS2 controller and receiver

2.6 Electrical Slip Ring

A slip ring is an electromechanical device used in situations when two parts are wired together, but need to be able to rotate individually. The slip ring transfer power or data from one fixed part to a rotating part. There are many types of slip rings. In this project, a capsule slip ring will be used. The wires from one side are connected to rings inside the capsule, and for the other side the wires go to brushes pushing against each ring. This allows the electrical connection to always be maintained under rotation. [12]

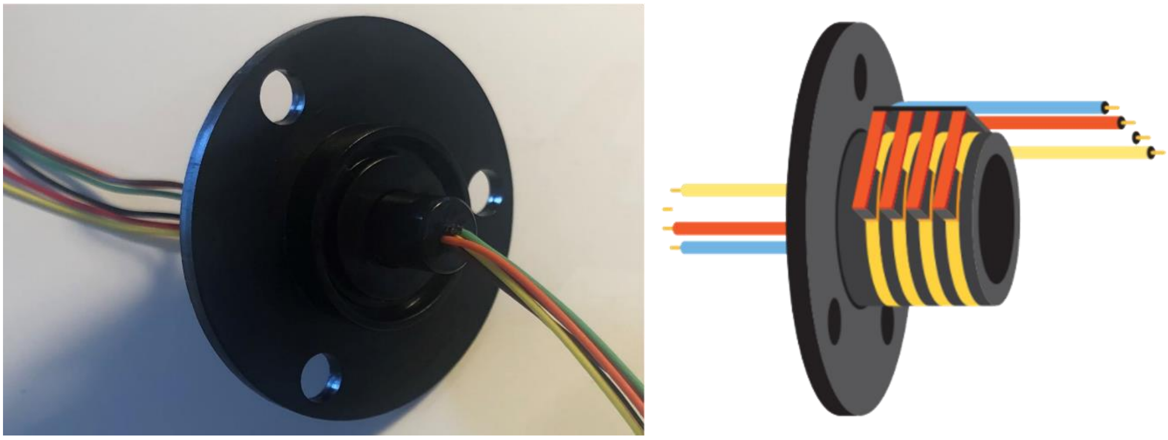


Figure 6: Slip ring and its inside [12]

2.7 Software

This section explains all the software used during the project.

2.7.1 Inventor

The software Inventor by Autodesk is used for designing all parts to be 3D-printed. It is a CAD software providing mechanical design, tools for product simulation and documentation, helping the user to simulate and visualize parts in 3D before manufacturing.
For more information about Inventor visit https://www.autodesk.eu/campaigns/inventor-2023-whats-new

2.7.2 PrusaSlicer

The PrusaSlicer software is a program developed by Prusa Research and is an open-source software made for creating the 3D-printing files. It takes the 3D object and breaks it down in layers to create the g-code for the 3D-printer. It also can detect where support material is needed on the model, and automatically adds this.
For more information about PrusaSlicer visit https://www.prusa3d.com/page/prusaslicer_424/

2.7.3 Arduino IDE

The Arduino IDE is an open-source software developed by Arduino. It is used for writing all code and uploading it to the Arduino board [4].
For more information about Arduino IDE visit https://www.arduino.cc/en/software

2.8 3D-printer

3D-printing is a manufacturing process included in the group additive manufacturing where, a three-dimensional model is created from a CAD model. The object is built by printing one layer at a time, from the bottom to the top. For this project, the technique used is called FFF/FDM(R)(fused filament fabrication/fused deposition modeling) and belongs to the category of material extrusion. To build the object, the printer heats the plastic filament and fuses it together with the underlying layer. For this project, the printer used is the Prusa i3 MK3S printer and the material used is PLA. PLA is a non-toxic, all-natural material classified as a thermoplastic polyester. It is based on renewable and organic sources, such as corn starch or sugar cane.

3 Method

This chapter explains the process of the project. In what order things were done and why. It includes the ideas behind the components and how it was all designed in CAD. It also includes the building of the circuit and the programming.

3.1 First Idea

The project started with brainstorming about what components would be needed, which parts had to be printed and how all the motion works. From this, a plan was made to draw all parts in CAD using Inventor and later 3D print them. When this is done the electrical components will be added. One question which surfaced was whether to use servo motors or something else. After researching, the decision was made to use linear actuators. These were chosen because the goal of the project is for the excavator to resemble a real one and the linear actuators both look similar and use the same linear motion. Other functions that the excavator needs to have, except the arm, are to rotate the upper body and being able to drive.

One requirement was also to have a switch to be able to turn on and off the excavator easily.

3.2 Electrical Components

When the first brainstorming session was done, the process of deciding how to control it started. The main idea of the project is to use a microcontroller connecting all the components. Arduino Uno was the first microcontroller that came to mind, simply because of previous knowledge in other projects. However, after research had been conducted about all parts needed, it was observed that the project would need more inputs/outputs than the Arduino Uno could provide and, therefore, the Arduino Uno was later changed to the Arduino Mega 2560 due to the number of outputs needed.

3.2.1 Controller

After choosing a microcontroller, a decision had to be made about how the user would control the vehicle. The first idea was to use a joystick of some sort. Two ideas were brought up, either it would be controlled by one bigger joystick with integrated buttons or two smaller joysticks. In the end, it was decided to use two smaller joysticks as in a real excavator, making it more realistic. After doing research, it was found that a wireless PS2 controller could be connected to an Arduino by using an imported library. This would suit the project perfectly, having both the two joysticks and spare buttons for extra functions. It was therefore decided to use a PS2 controller as main control.

3.2.2 Motion of Arm

For the movements of the arm, the choice had fallen on micro linear actuators which have the same motion as hydraulic cylinders. For this project, a linear actuator with attachment holes on both ends was needed for the movements to work correctly. The next step was to power them. They need to be supplied with 12 and 6 volts, and because the Arduino can only deliver 5V, an external power source had to be connected. To make sure the 12V circuit never crosses the Arduino circuit, the first idea was to use some kind of relay. This idea was quickly changed to instead use a motor driver with an included H-bridge. This decision was taken because the linear actuators are controlled by changing the direction of the current. Another crucial factor is controlling the speed of the motor, which is something you cannot do using only a relay.

3.2.3 Driving

The excavator will be driven using two motors together with caterpillar tracks, also known as continuous tracks. The two motors chosen for the project are DC-motors with integrated gearboxes, called TT motors. These motors distribute the torque to two shafts making it possible to drive two wheels with one motor. In this project, two gears will be connected, one on either side, for driving the caterpillar track. For controlling the speed and direction of the motors, the L298N motor driver was chosen.

The other parts needed for driving will be 3D printed, including the track links, gears, and attachments to put everything together.

3.2.4 Rotation

For mounting the main upper body to the base with the tracks, something was needed for the main body to be able to rotate freely. The first thought was to use one external gear together with a DC motor, with a smaller gear placed on the base plate. To find a gear suitable and affordable for this project was difficult, and new ideas had to be thought of. After doing more research, a rotational plate was discovered. The first one chosen had reviews which said that the force had to be exactly centered for the rotation to work well. From this, the idea to use ball rollers was brought to light. The balls would even out the load from the body and help with the rotation. After conducting more research of rotational plates, another similar option was found. A rotating plate made for rotating chairs. This could then take all the force even if it were a little bit unevenly distributed. This new plate replaced the former plate and the ball rollers. This choice was later changed when new problems arose.

The next problem to solve was how to connect the cables between the motors on the base and the microcontroller on the body. When the excavator rotates, the cables would be twisted. Even though this could work for one turn or so, the goal is to be able to spin an infinite number of turns. To solve this, an electrical part which is made of two parts that could rotate individually was needed. Research was made on what kind of joints exist on the market, and a slip ring was chosen which had six cables. This component allows the four wires to the motors to be connected and the two wires going to the rotational motor.

To figure out how the rotation will be driven was one of the trickier problems to solve. The thought was to either use a servo motor or a DC motor. Both types of motors would have to have a high torque and the speed is not too important here. Using a servo motor for rotation, the motor would, as mentioned, need a high torque but also it needs to be continuous. Most servos can only rotate between a fixed angle. To be able to rotate a complete lap, the servo must be built in a unique way. These two factors, torque and rotation, would increase the price significantly. The other option to use a DC motor is more affordable, and will do the job equally as good. DC motors are more common with high torque, and infinite rotation comes as standard. What needs to be considered when using a DC motor is how you would control it. If the motor is below 6V, then it can be powered straight from the Arduino. For this task the voltage would have to be higher to achieve the torque needed. Therefore, the motor driver mentioned earlier (L298N) would be needed. In the end, it was decided to use a DC-motor with the motor driver with the justification that it is more affordable and easier to control using a H-bridge.

When thinking about the assembly of all the rotational components, another problem was found. It would not matter if a DC motor or a servo motor was used, either way the cables would twist around the shaft when the motor is placed in the middle. The slip ring would untwist the cables in relation to top and bottom, but because the motor's shaft would have to be centered, the cables would still twist around it. Now, the thought was to go back to the original idea to use external or internal gears together with a smaller gear, all placed on the base. As mentioned earlier, it was not easy to find a suitable gear. New research on how radio-controlled vehicles with rotation works were made. Looking into if there exist spare parts or similar on the market for this kind of rotation. A kit was found combining the gear and a plate for rotation. To this kit, a motor fitted with a small gear was also found. The decision fell on using these, see *Figure 7*.

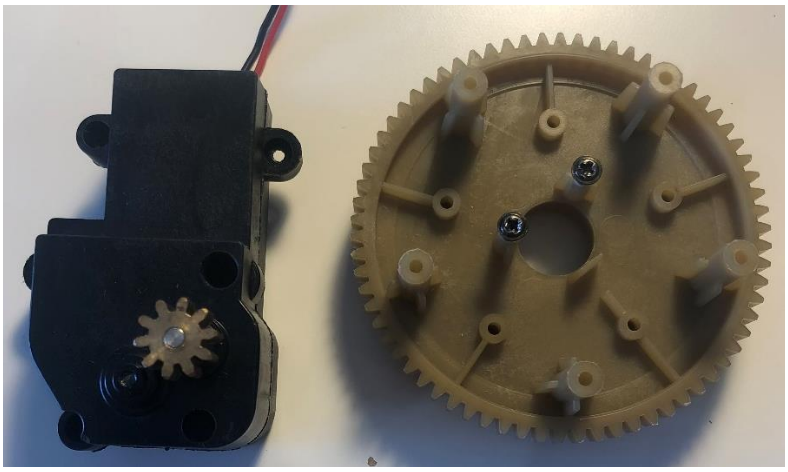


Figure 7: Rotational motor and gear

3.2.5 Power Source

For the external power source needed to power all motors and linear actuators of 12V the idea was to find a small, compact, and rechargeable battery. With a first search on the internet, it was clear the options were not that many. Batteries with 12V can, for example, be found in motorcycles, but these are too big and heavy. The next idea was to search among the parts for radio-controlled vehicles. The offers here are limited, with most batteries having a voltage of maximum 11.1V. One battery being able to supply 12V was found together with a charger. One thing to keep in mind is that the motor driver has a voltage drop of about 2V, which means that the linear actuators will never be able to drive at full speed. Because it was near impossible to find any bigger battery for a reasonable price and full speed is not important in this case, this was considered acceptable. Later, a new sort of battery was discovered, the 3.7V 18650 Li-ion batteries. By connecting three of these in series, a voltage of 11.1V could be created which is slightly under 12V. Connecting four of these would give a voltage of 14.8V, and although the motor drivers can manage this, the on-board 5V regulator would have to be disconnected. Because 14.8 volts is not necessary and 11.1 will work fine, three batteries will be used to save space. They are placed in a holder connecting them all together, see *Figure* 8. To distribute the power to every component, wago clamps with five inputs are used.

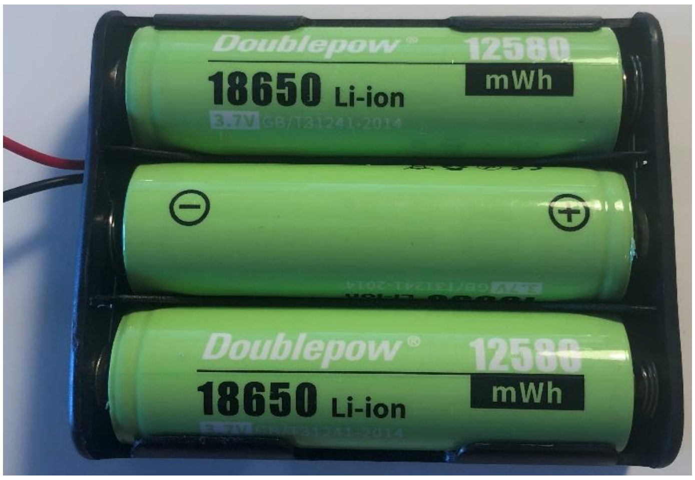


Figure 8: 18650 Li-ion batteries in the holder connected in series

3.2.6 Assembly of Circuit

Each motor driver can power two motors and, therefore, three are needed in total. One for the two motors driving the excavator, one for the rotational motor together with one of the linear actuators, and the last one for the last two linear actuators. With this information, a new problem occurred. With the original idea to use an Arduino Uno this would not be possible. Each motor driver needs six inputs making a total of 18, and the Arduino Uno only has 14 digital outputs. Furthermore, the PS2 controller also needs four inputs. To solve this a decision to upgrade to the bigger Arduino 2560 MEGA was taken. This microcontroller has plenty of inputs/outputs and will be able to manage all cables and more. For the complete circuit see *Appendix D. Electrical Circuit*

3.3 Design With CAD

When all electrical components were chosen, the process of drawing all the parts in CAD began. The plan was to start with the arm because this was to be the most complicated part with a lot of movable joints. At this point in time, the electrical components had not yet arrived and, therefore, it was only possible to approximate all the measures of the components. All parts were drawn in the program Inventor by Autodesk. This program was chosen because of previous experience and a lot of time spent on getting to know all the functions in it. This would reduce the time taken to draw all parts.

Because all parts drawn in CAD would be 3D printed, a lot of thought went into how they could be designed in a way that makes it easy to print and not having the need for support materials. Also, to try to reduce the weight of the parts and the time for the printing itself. From previous experience, it was known that support material could be hard to get off, and for important holes it is good to have margins in case it is not possible to get rid of all the support material used. It was also known that the use of margins in general would be needed. Even if the parts fit perfectly together in the CAD program, in reality they do not. These margins were introduced in every place to the various parts that would connect to each other and for all the holes and pins. To know how big the margins would have to be, test parts with holes and pins in varied sizes were printed. From this, it was discovered that for all holes not using support material, the size of the hole can be the same size as the screw. This would allow the screw to be perfectly screwed in place in the hole.

During the entire process of creating every part, an assembly file was kept open to see if something looks off when adding the parts together. This made it possible to see what changes had to be made in real time, instead of putting it all together at the end and finding out a lot of adjustments must be made.

When thinking about the design, the first plan was to use the design of a mini excavator, but this design is very compact, and when considering how all parts are to be fitted, a conclusion was made that it would not be possible. For this reason, the design was changed to resemble a bigger excavator. This also made it possible to distribute the parts over the body and, therefore, distribute the weight to create a counterweight from the components to act against the force from the arm.

3.3.1 Arm and Bucket

The arm of an excavator consists of two pieces. The first, closest to the body is called the boom, and the second is called the stick. See below for the names of various parts on the arm.

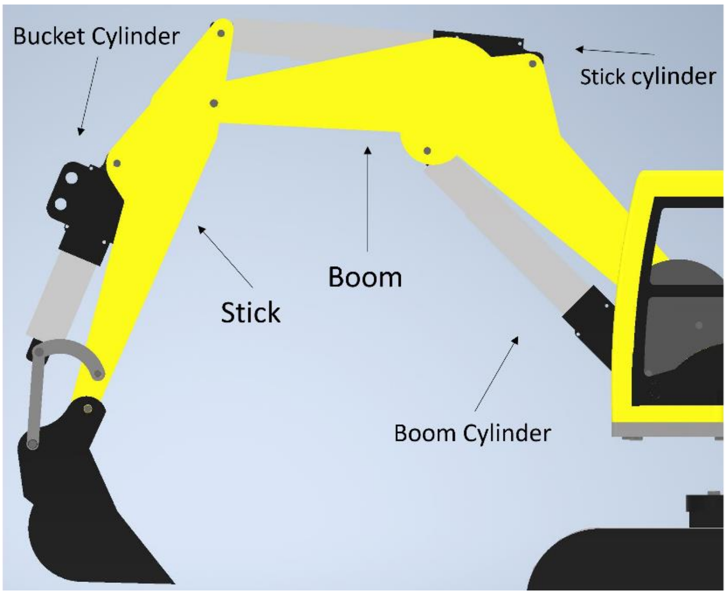


Figure 9: Name of parts on the arm

The first part drawn was the boom. Images from the internet were used as reference on how it would look. The size of the arm was adapted to the size of the linear actuators used. Instead of having one thick arm, it will be separated into two thinner parts. This reduces the overall weight of the arm and the printing time. The cylinder can then be attached between these parts. On real excavators, the boom cylinder is sometimes attached to the side of the arm, sometimes on top of it, and sometimes underneath. For this project , it will be attached underneath, in the center of where the boom curves. There was no special reason for this other than design preferences. All other necessary holes were added, such as the joints for the cylinders, the joint between boom and body and the joint between boom and stick. On one of the boom arms traces are added for the cables coming from the linear actuators to go through.

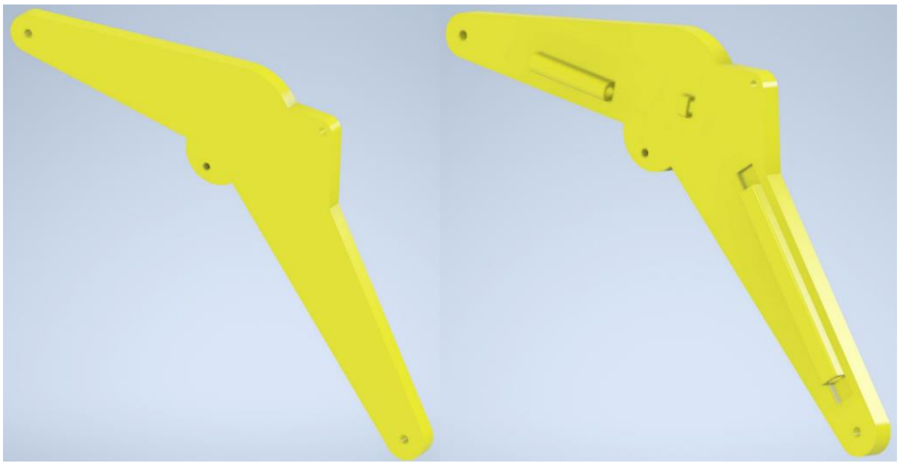


Figure 10: Boom arms, right boom with cable trace

Second part designed was the stick. Like the boom, the design of the stick was also inspired from real excavators. It was also separated into two thinner parts at first, but when later assembling every part in CAD, there was not much space left between the parts. Instead, one thicker part was used. By changing to one thicker part and not having any space between, some modifications had to be made. Both the stick and bucket cylinder must be able to move freely. For the connection to the stick cylinder, material was taken away in the middle of the part. The same goes for the attachment of the bucket cylinder, where material was taken away to make space for the cylinder to move. At last, all the holes were added for all joints. A hole running through the stick is created allowing the cables coming from the bucket cylinder to go down to the boom easily.

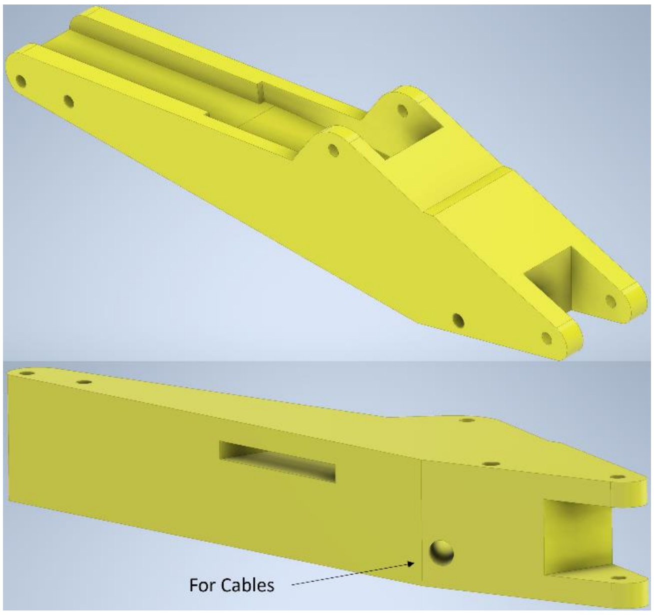


Figure 11: The Stick

When all parts for the arm were ready, the design of the bucket started. For an excavator, there are three typical buckets that are used. For this excavator it was decided to make the digging bucket ,because this is the most common. First, the bucket itself was created, and lastly the attachment connecting the bucket and the stick. For the bucket to work smoothly together with the bucket cylinder and the stick, two more parts are needed. These are the two links which create a distance between the cylinder and the stick, and make the bucket turn. By looking at different designs, it was clear two options were frequently used. Either to use two straight links, or one curved and one straight. Both options would work for this project, and the option with one curved, tipping link, and one straight, bucket link, was chosen mainly because of the appearance. The measurements of the two links were first approximated and later evaluated together with the other parts and adjusted to appropriate sizes for creating a smooth digging motion.

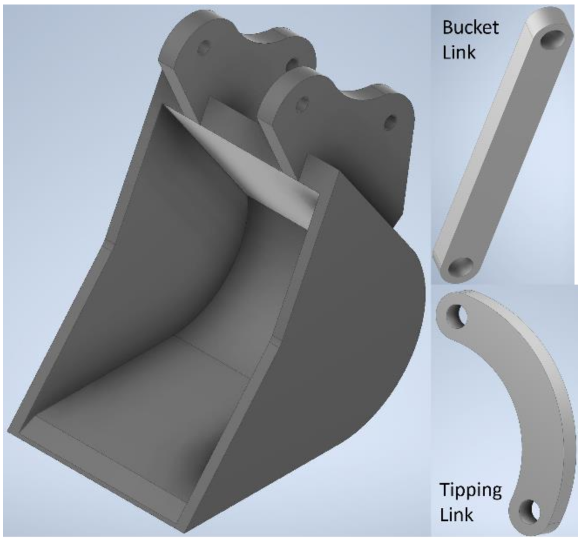


Figure 12: Bucket, bucket link and tipping link

To facilitate adjustments to all parts and to see how they will work together, basic cylinders were created in varied sizes resembling the real ones. From this, it was discovered that the original plan with a small bucket cylinder was not possible because of the small stroke. One with a larger stroke would be needed. After these changes, it was possible to adjust the placements of all the holes for the different joints.

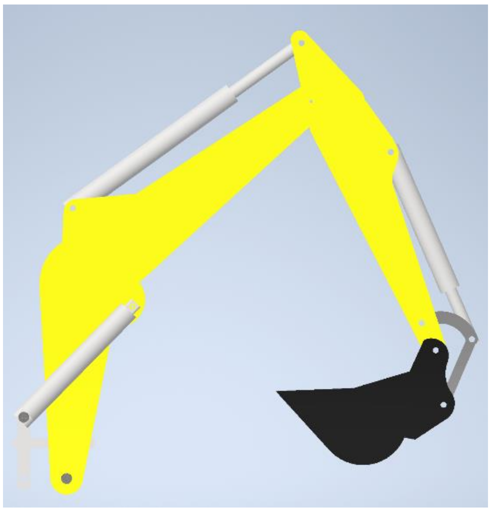


Figure 13: Test of assembly of the arm

Lastly, when all parts for the arm was completed, it was time to figure out how to attach it to the excavators' body. Some different variants were thought of. These included connecting it straight to the cabin to resemble a real excavator, using a standalone part only for the attachment, and creating the base plate for the body with an integrated spot for the arm. The final decision fell on making a standalone part that would first be attached to a baseplate, and after the arm could be installed in it. This decision was made in case the printing material would not be able to withstand the forces. It would be easy to print only this part in a harder material, for example, ABS plastic.

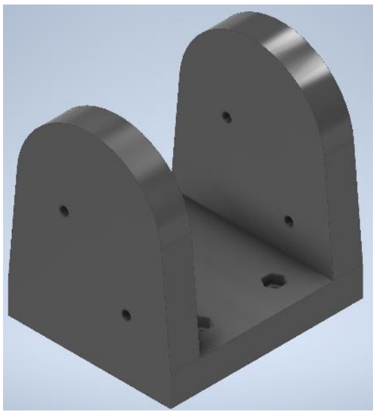


Figure 14: Attachment for the arm

3.3.2 Cabin

Now, the process of creating the cabin started. This part of the excavator has no functionality, and it is only made for design purposes. The goal for the cabin was to have some details, but nothing too advanced. One important thing is to have a door that can be opened. Other features included having a chair and one piece to resemble a couple of indicators. For the size of the cabin, no measures existed. Instead, it was built in a size that would look good relative to the rest of the body.

The first thing created was the cabin itself. To facilitate the 3D-printing process, and to not have a lot of support material to clean, the roof was created as a separate part. The main cabin body was created, and doors and windows were cut out. Holes to attach the cabin to the excavator, raised parts for the chair and display, and support for the roof were created. The roof was then created with the same shape as the outlines of the cabin, and pins were added for attachment. On the side with the door, an attachment was created with the idea that the door would only rest on this attachment on the bottom. The cabin would have a pin and the door would have a hole. In the pin, a 90-degree hole was created. In case the door was going to be lose, a pin could be inserted in this hole, preventing the door from falling off. Later, one more pin was added in the upper part of the door to prevent the door from falling over. The door was made using the outlines of the door hole in the cabin, and then scaled down a bit to have some margins around.

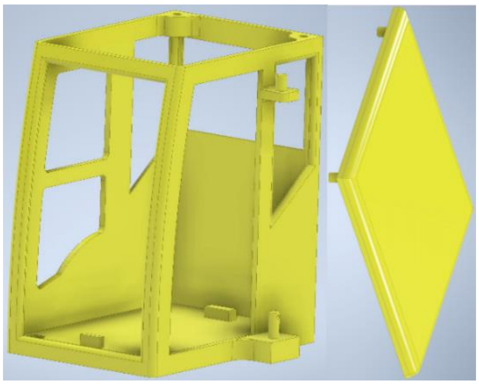


Figure 15: Cabin body and roof

For the chair, the design is a simple one including the two joysticks. To not have to print with support material, elbow rests were made for the joysticks to attach to. Next the part with a display was created including some indicators. Both these parts have holes underneath attaching to the pins on the cabin floor.

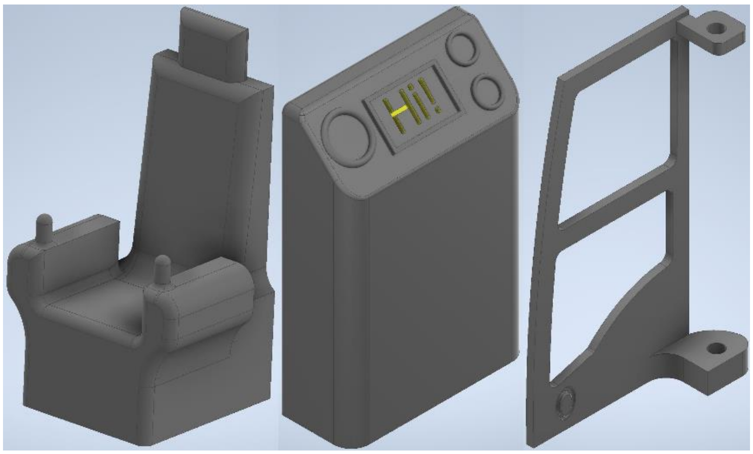


Figure 16: Cabin chair, display, and door

Lastly, all the parts were assembled and placed next to the arm to see how it would look together. See below for the finished cabin.

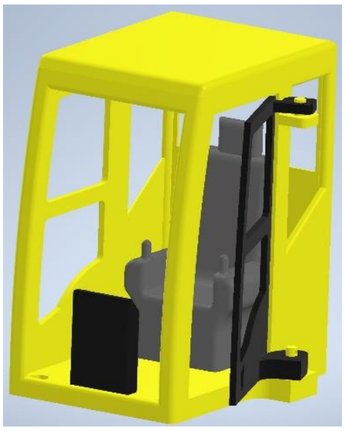


Figure 17: Assembled cabin in CAD

3.3.3 Electrical Parts

When the electrical parts arrived, they were measured and drawn in CAD to easily visualize all the parts together. Two parts were downloaded as ready-made parts and slightly modified to correct the most important dimensions. The parts that were downloaded are the motor driver [9] and the Arduino [10]. The main reason for downloading these parts is for their appearance in the CAD program, but also to get the dimension. These were downloaded before the real parts arrived. Drawing all electrical parts in Inventor facilitate the placement of every part. It makes it possible to see which part can fit where, and if there are any collisions between parts or the cover. See below for process of deciding placement of parts.

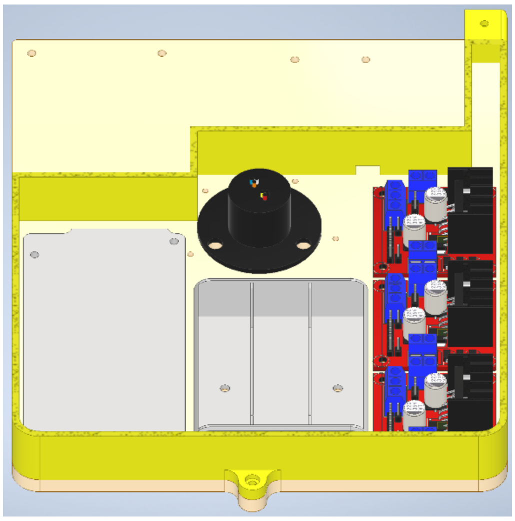


Figure 18: Testing placement of parts

3.3.4 Plate for Body

Now when the arm and the cabin was done, it was time to start thinking about the plate for the body. The plate is rectangular with all holes needed to fasten all the parts. This includes the holes for the cabin, the arm attachment, slip ring, rotary gear, and the battery holder. Two rectangular holes were also created in the back of the board to allow for a table to be attached over the battery holder. This table is used in case more counterweight is needed, then it can be placed on top of it. In the end, this part was not used.

For the Arduino and the motor drivers, pins were created instead of holes, making it easy to attach and detach them when changes need to be made with the cables.

Extra holes were placed next to the slip ring, and in the front, close to the arm. These are used for attaching cable ties and making cable management easier.

The last thing for the upper body of the excavator was to build a cover to cover up all the electronics and give it some finishing touches. Attachments for this cover were created on the plate, one in the front corner and one centered at the back. See *Figure 19* for completed plate.

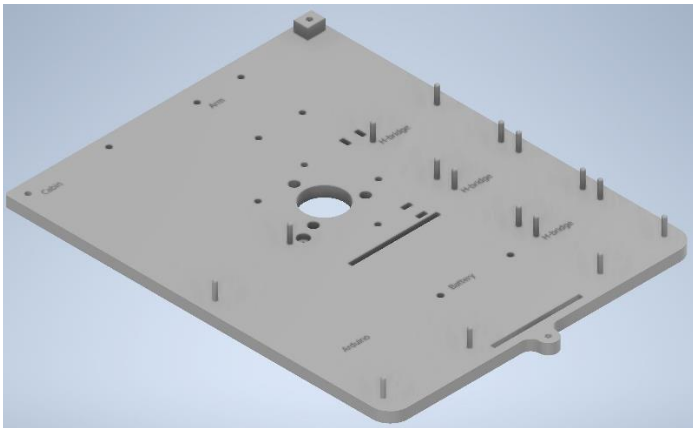


Figure 19: Plate for body

3.3.5 Cover

The main function of the cover is to hide all the electronics and to, therefore, improve the overall appearance of the excavator. The cover is the same size as the base plate, with cutouts for the cabin and the arm. The electrical switch for the excavator is integrated on the side, and on the top, a hole is made for the receiver to be placed. The receiver is placed outside the cover to minimize disturbance of the signals and, therefore, maximize maximum distance it can be driven from. Another cover to hide the receiver is created, resembling an engine cover. It also helps keeping the receiver in place. Lastly, an exhaust pipe is added for design.

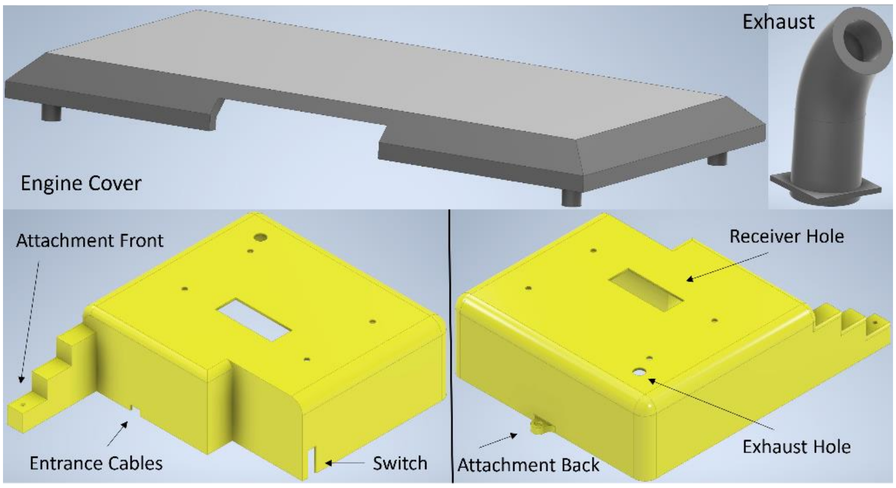


Figure 20: Cover, engine cover and exhaust

3.3.6 Base Plate

The base plate is the plate connecting the upper body with the lower driving body. It includes the holes for the rotational gear, the electrical slip ring, and connections for the tracks. The motor for driving the rotation is also fitted on this, placed in a pocket so the gear line up with the big gear. The tracks will be placed under the plate, and to make sure they will not touch, raised parts were created at the attachment. The shape of the plate started as a rectangle and later parts were cut out to minimize use of material and therefore also weight. Lastly, extra holes were added for cable management. An extra piece was created to work as a lock for the rotational gear. This piece is a simple cylinder with hatches on, and is to be placed through the rotational gear and go down to the base plate.

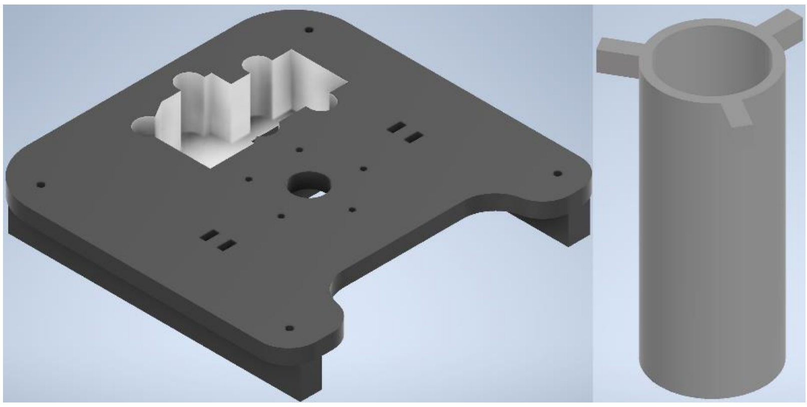


Figure 21: Base plate and rotational gear lock

3.3.7 Tracks

The driving tracks are made up of several parts. The design is based on a design existing on Thingiverse, made by the user *nahueltaibo,* and modified to fit the project. The assembly is made up with the track links and two kinds of gears, one driving and one for support at the other end. These parts are all taken straight from Thingiverse [11].

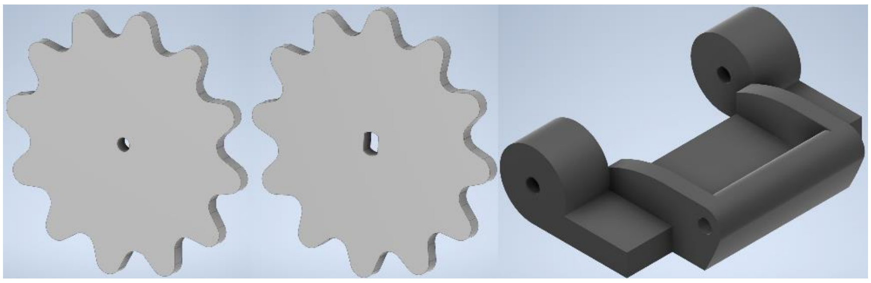


Figure 22: Support gear, driving gear and link

The other parts include the chassis between the gears holding the motor. This part was completely redesigned but using the same type of attachment for the motor. To lock the gears from not falling off outwards, a simple flat part was created and attached on the outside using screws. This lock is installed using the hole for the screw in the back, and on the other side, attached with a screw to the motor. Because the motor screws are small, an extra piece had to be made. This piece is a simple washer allowing the small screw in the motor to also fasten the lock. For the inside, a part is needed to connect the tracks to the base plate. The first idea was to print angle brackets, which later turned into a piece covering the whole inside and included a "shelf" for the base to be connected to. This part now stops the gears from falling off and allows the complete track to be attached to the base plate. Four distances were created and placed between the main chassis and this piece, to allow the gears to rotate freely, and separate the part from the track. Later, during the assembly of the track, the links sometimes jumped off the gears which led to the redesign of some parts. The "lock" holding the outer gears was enlarged creating a wall for the links to prevent them from moving in this direction. On the other side, the piece connecting the tracks to the plate was modified in the same way, creating walls preventing the links from moving sideways.

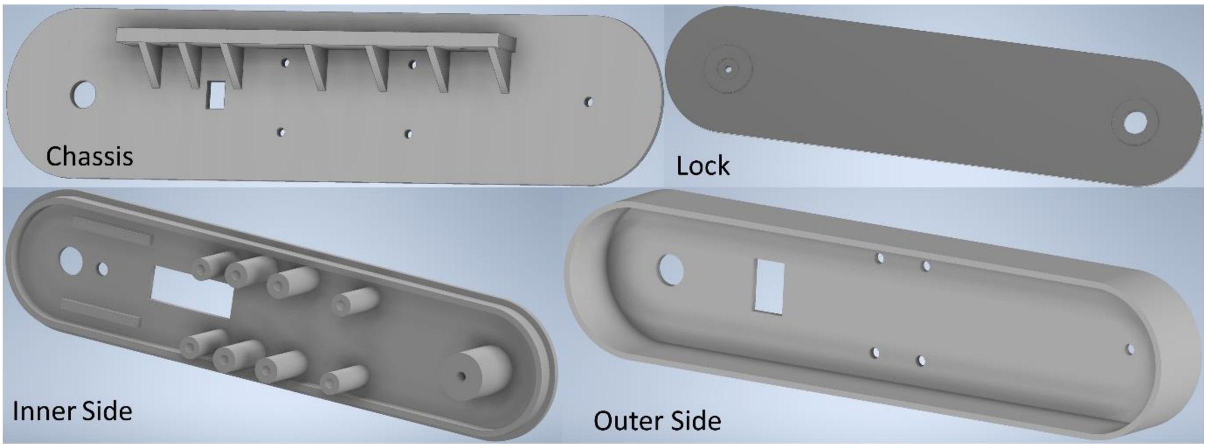


Figure 23: Chassis, lock, inner side, outer side

3.3.8 Assembly

When all parts were done, the assembly was completed in CAD. This made it possible to see that everything fits together or if there are parts or positions that needs to be changed. See *Figure 24* for the completed assembly in Inventor.

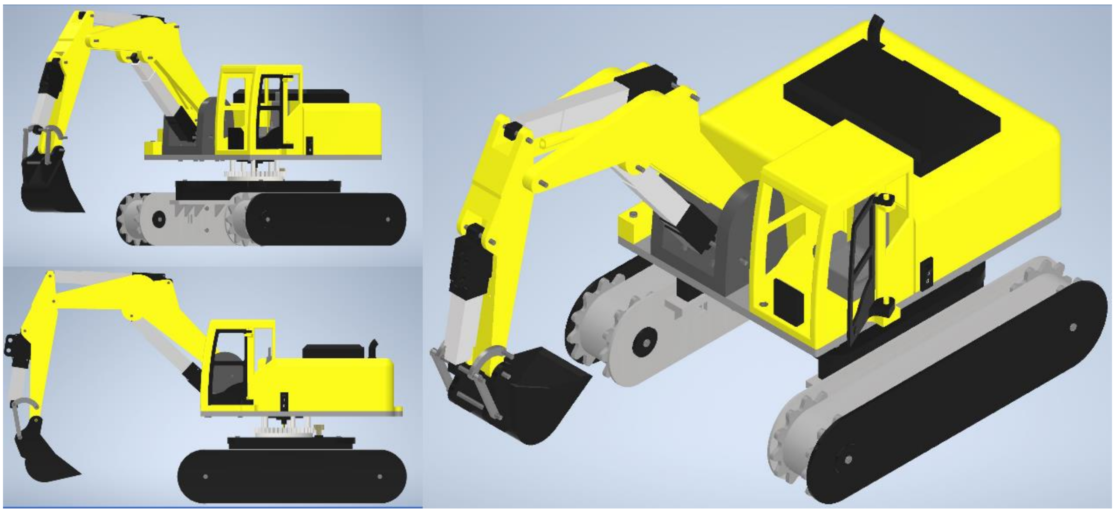


Figure 24: Final assembly in CAD

3.4 Programming

The excavator needs to have the basic functionalities as a real one has. This includes the two movements of the arm, the bucket and rotation of the upper body. It also needs to be able to drive.

The first thing in the code is to import the library for the PS2 controller [13]. After this, all the necessary variables and pins are defined followed by the functions. In this program, functions are used for the driving.

In the setup function, all pins are set as either inputs or outputs depending on their functionalities. The program is then checking if the PS2 controller has been connected properly. If not, it skips the rest of the code. If it is connected properly, it continues with the code and the excavator is ready to run.

The first thing happening in the main loop is the microcontroller reading the state of the two joysticks, and saves these values. These values are later used in if-statements checking which operation is requested. For the complete code, see A*ppendix B. Arduino Code.*

3.4.1 Joysticks

The two joysticks are used for controlling the two motions of the arm (the boom and stick), the bucket, and the rotation. The joysticks work as linear potentiometers in the X- and Y direction, sending values between 0-255 depending on its position. The microcontroller reads the values from the PS2 controllers' joysticks and converts these to an interval between -1023 to 1023, this is to easily see in which direction the motor should go and by how much. Furthermore, this allows for a small interval where nothing happens to make sure the motors do not spin back and forth when small unintentional motions are applied to the joystick. The microcontroller then sends these values as a PWM signal to the motor drivers to control the linear actuators and motors. See *Figure 26* for the flowchart.

3.4.2 Driving

For driving the excavator, the R1, R2, L1, L2 buttons of the PS2 controller are used, see *Figure 25* for map of buttons. The R1 and L1 buttons are used for driving each motor backward and the R2 and L2 for driving forward. With different combinations of these buttons, the excavator can drive forward, backward and by combining one forward button with one backward button it will turn. To code this, the Arduino checks which buttons are pressed and then sends the direction and speeds to the corresponding motor driver. A button is set as True when pressed, and to False when released, making it possible to save the values as Boolean and read these in if-statements. To program like this, instead of just using four buttons for all directions, allows the user to determine how much each track will move and can choose how much to turn, instead of having a predetermined value. See *Figure 26* for the flowchart.

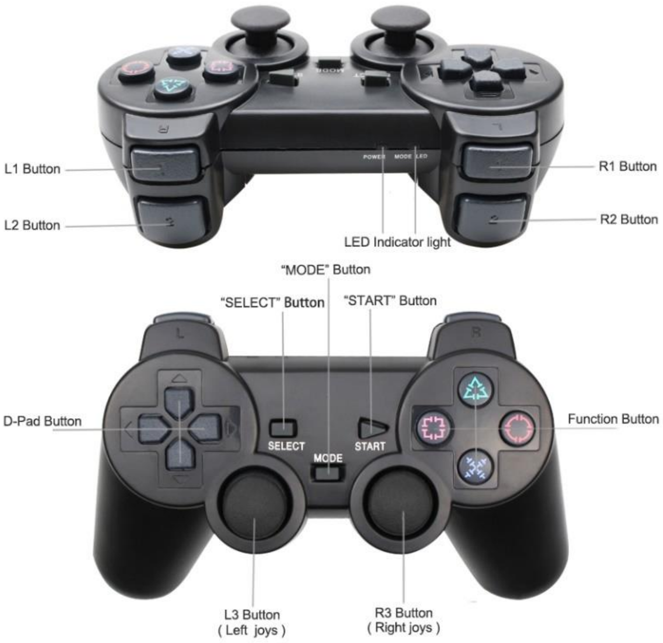


Figure 25: PS2 controller with named buttons [16]

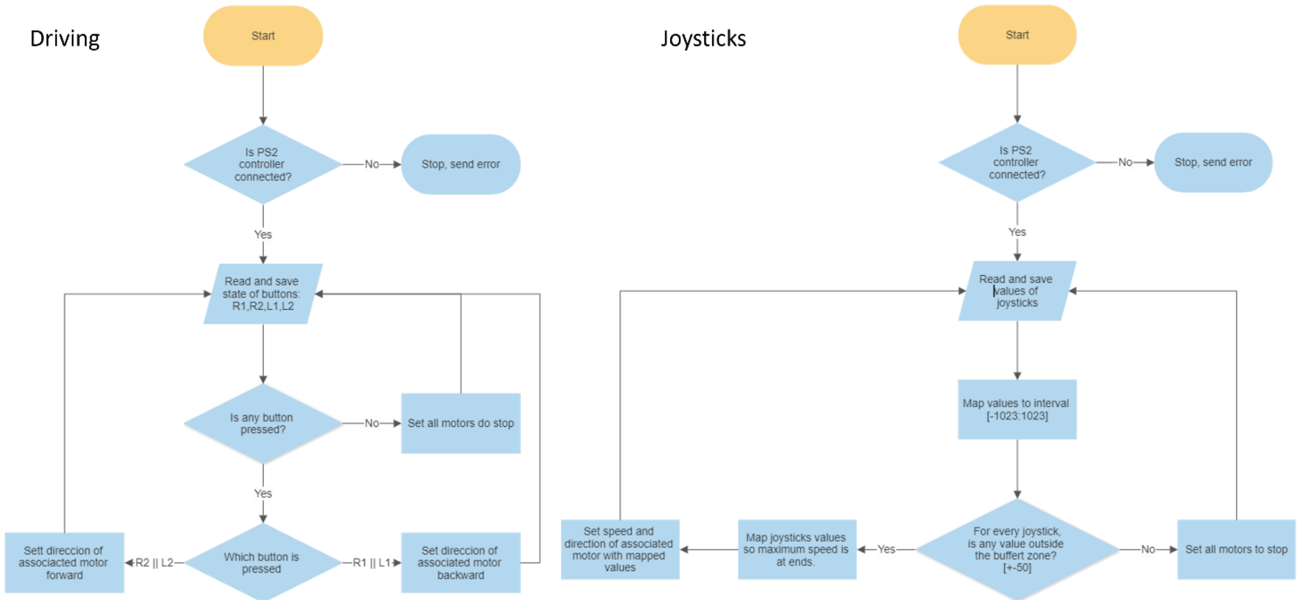

Driving
Start
Is PS2 controller connected?
No
Stop, send error
Yes
Read and save state of buttons: R1,R2,L1,L2
Is any button pressed?
No
Set all motors do stop
Yes
Which button is pressed
R2 || L2
Sett direccion of associacted motor forward
R1 || L1
Set direccion of associated motor backward
Joysticks
Start
Is PS2 controller connected?
No
Stop, send error
Yes
Read and save values of joysticks
Map values to interval [-1023:1023]
For every joystick, is any value outside the buffert zone? [+-50]
No
Set all motors to stop
Yes
Map joysticks values so maximum speed is at ends.
Set speed and direction of associated motor with mapped values

Figure 26: Flowchart of code, driving and joystick

4 Assembly

This chapter explains how the final excavator was built. It describes the assembly process of all the parts and the integration with the electrical parts.

4.1 Tracks

The first element built was the tracks, as they are independent from the rest of the excavator. The motor was placed in its chassis and the gears were added. The lock for the gear was secured and the plate for mounting the tracks to the body was attached together with the distances. For the track itself, all the links were put together using raw filament as pins and then attached to the rest of the assembly. The ends of the filament were melted to stop the pins from falling out.

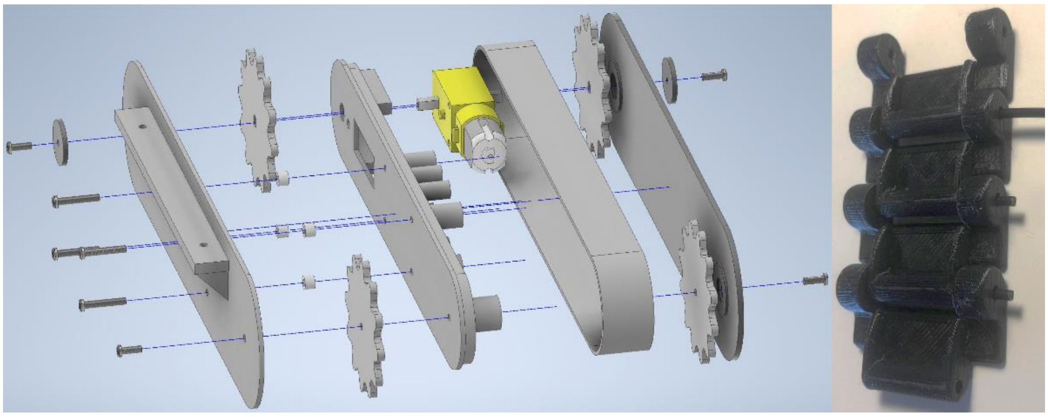


Figure 27: Exploded view of assembled track and assembly of links

4.2 Arm

Next piece that was built was the arm. The linear actuators were placed in position together with the printed boom and stick. The cables were threaded through the cable paths. The bucket was attached at the end of the stick and the links were attached. To make sure the linear actuators do not accidentally move to a diagonal position, nuts are threaded on to the screws, one on each side of the actuators. These nuts are placed on both ends of the actuators. To make sure they do not get stuck in an odd position, margins are introduced between the nut and actuator to allow for smaller movements.

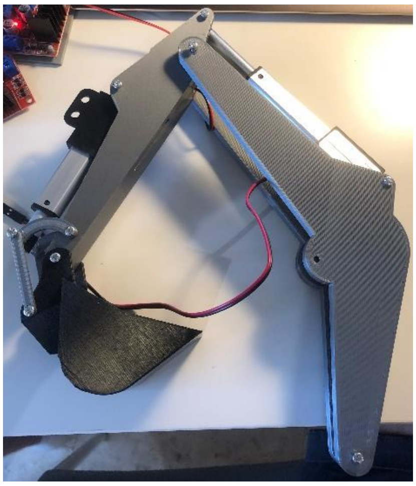


Figure 28: Assembled arm with bucket

4.3 Cabin

The cabin was built around the main body. First, the chair and the display were attached to the pins on the floor of the body. Next, the roof was attached with the pins, followed by the door. To secure the door from not falling off, a piece of filament was inserted working as a locking pin.

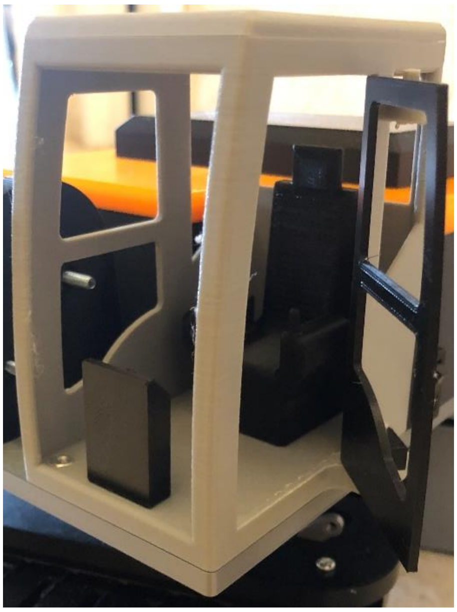


Figure 29: Assembled cabin

4.4 Base Plates

The rotational motor was placed in its pocket in the base plate followed by the attachment of the rotational gear. The plate was then connected to the two track assemblies, creating the foundation of the excavator. Next, the lock for the rotational gear was inserted in the center hole before the plate for the body was attached to the rotational gear, connecting the lower part with the main body. The slip ring was secured in place, and the cables threaded through to the underside of the base plate. Lastly, the holder for the arm was mounted on the plate.

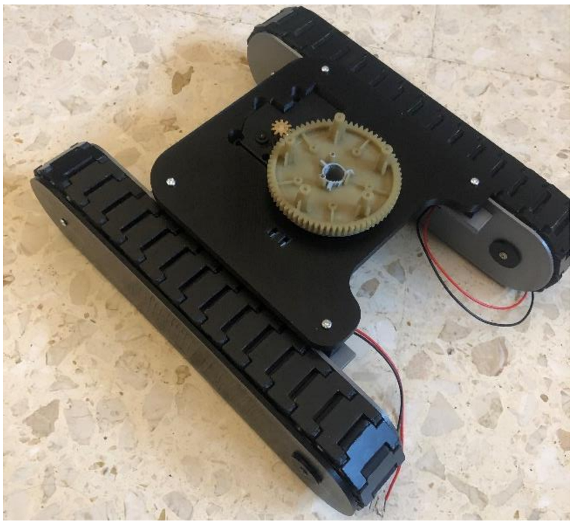


Figure 30: Assembled base plate including tracks, rotational gear + lock, and rotational motor

4.5 Final Assembly

Now all the smaller assemblies were joined together. First, the cabin was mounted on the plate, followed by the arm to its holder. Two nuts are threaded on to the screw holding the boom arms. Their function is to push the booms towards the sides, reducing the margins between the boom and the attachment, making the whole arm more rigid. Then, the motor drivers, Arduino and batteries were installed. Now all wiring for the linear actuators, motors for driving and the rotational motor were done, including soldering all wires that needed to be. Cable ties are used to tidy up the cables.

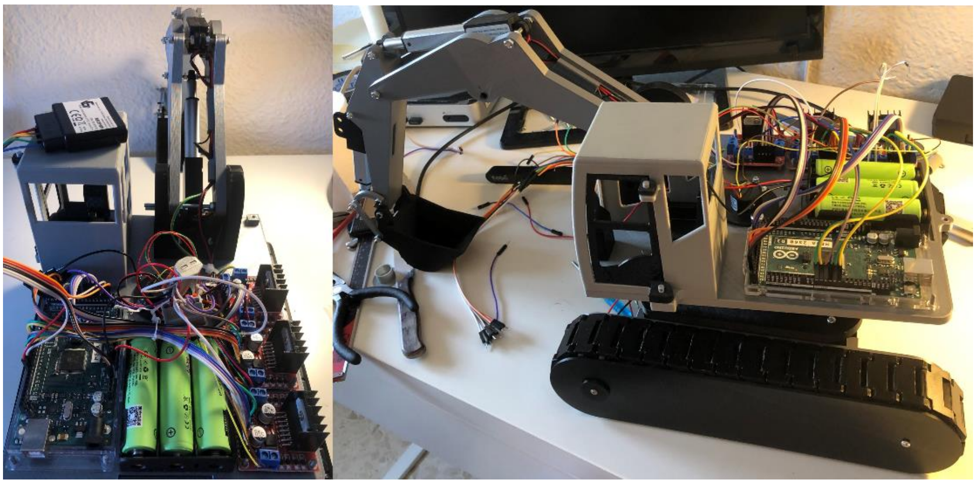


Figure 31:Excavator assembled without cover

The last part is to connect the cover. First, the receiver was connected to the Arduino and pulled through the cutout in the cover. The switch was inserted in the cover and connected to the circuit. The cover was then placed in position with the screws. The engine cover and exhaust pipe were added at the end.

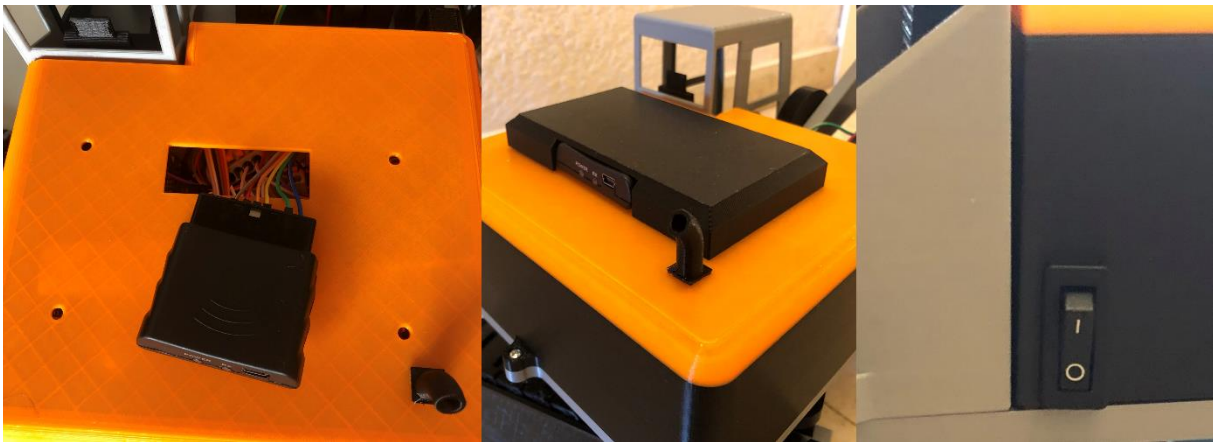


Figure 32: Installation of cover

For the complete electric circuit, see *Appendix D. Electrical Circuit.*

5 Results

The final product is working as intended. The 3D printed parts are assembled as planned in CAD and allow for all the motions to be performed.

The arm of the excavator works smoothly and can reach about 30 cm away from the body. A maximum height of the bucket at 40 cm, and a maximum digging depth of 26 cm, see *Figure 33*. All joints and parts work together as intended, exactly as the CAD model. The linear actuators move freely and can reach both the end and starting point without any problem.

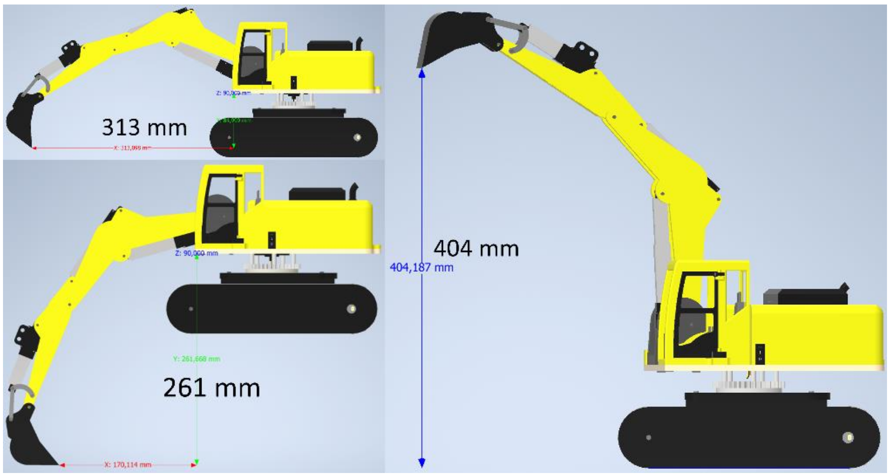


Figure 33: Maximum digging lengths

The final testing of the product includes assessing every motion for themselves and how multiple motions work together.

The first phase of testing is evaluating the motions of the arm. Starting with the boom cylinder, making sure it can lift the arm and also take its weight when the arm is lowered and not letting the arm fall down. This was a success and the linear actuator could manage the weight of the arm both ways. When the arm is fully stretched out, the actuator slows down a lot due to the moment created at the tip.

Second motion of the arm is the stick. This part work flawless with a perfect maximum speed and no difficulties moving the stick and bucket.

Last motion is the bucket. The links work perfect and help to create a smooth digging motion. The actuator has no problem moving the bucket back and forth.

For all motions of the arm, both the start and ending point can be reached without collision to other parts. Controlling the speed of the different actuators works great, but due to the maximum speed of some of the actuators, it is barely noticeable. For example, the buckets actuator is slow and, therefore, no change in speed can be seen.

The final test of the arm was to combine the motions of every actuator. This also turned out to be a success, but worth mentioning is that the movements became slower due to all actuators being used at the same time. Using two actuators at the same time works perfect, adding the third slows the movements down.

The testing of the rotation includes testing to turn both ways, controlling the speed, making sure it can turn for an infinite time. The excavator passed all these criteria without any difficulties. The slip ring works exactly as expected making sure the cables do not twist and allows for continuous rotation.

Test of driving includes evaluating each track for itself and then testing them together. By them self, the tracks are too weak to drive the excavator and, therefore, it is not possible to turn the excavator. When tracks are working together the excavator can move and, therefore, making it possible to drive forward and reverse.

Testing multiple parts of the whole excavator included: moving the arm at the same time as rotating, rotating and driving, and all three sections together. The results from this were over expectations. Every part of this test worked, though performing many movements at the same time slows down all individual movements. This is acceptable and not a big deal considering the same thing happens in real excavators.

The maximum distance of stable and reliable connection is at 15 meters in good conditions, meaning no obstacles between the controller and the excavator. Going over 15 meter the connection gets unreliable, sometimes working and sometimes not. The time for the controller to connect to the excavator is near to instant.

Dimensions of the body is 230 x 187 x 110 mm, for all dimensions see *Appendix C. Dimensions*.

The total weight of the excavator is 1621 grams.

For the lift capacity, in its current condition it cannot lift anything. The reason for this is of how the gear connecting the lower body with the upper is constructed.

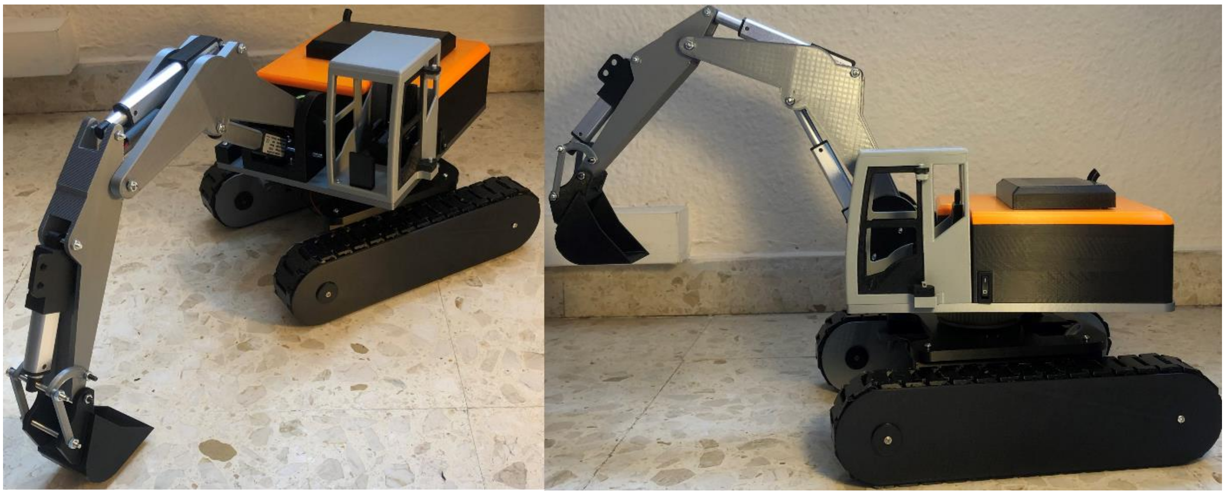


Figure 34: Finished excavator

6 Discussion

6.1 Results

The final product meets the requirements of the scope and, therefore, a conclusion can be drawn that it is possible to build a remote-controlled excavator using 3D printed parts. All essential movements that a real excavator has, can be found on this model. The overall design resembles a real excavator and every part has the right proportion compared to the others.

The arms movements work in the same way as a real excavator, and every part moves smoothly. The linear actuators are strong enough to lift the hole arm, and there is no problem for the attachment between the arm and the body managing the weight.

The track assembly works great alone. The two sides keep the links in place and with the margins between the links and sides, it can move freely without any additional friction. The inner side, connecting the track to the excavator, can manage the weight of the excavator without any problems.

To control the excavator with the PS2 controller is both simple and resemble driving a real excavator. The two joysticks are precise and to change the speed of any motor is easily done. The buttons for driving works perfectly, although the motors are too weak to individually drive the excavator. With a maximum range of 15 meters stable connection, it is a success using the PS2 controller as the main controller.

6.2 Future Improvements

As mentioned, every movement except turning works as expected. Future improvements that can be made includes the arm, the lift capacity, and the driving.

Starting with the arm, due to the variation of speed and maximum force of the linear actuators, some parts move faster than others. For example, the boom cylinder has a maximum speed of 15 mm/s and maximum force of 60N, the stick cylinder 7 mm/s with 128N and the bucket cylinder 4 mm/s with 150N. As a result, the stick cylinder moves the fastest and the bucket cylinder the slowest, making it difficult to create smooth digging motions. This is easily fixed by using linear actuators with the same speed and force. The reason it is not used in this project is because of problems receiving parts in time, and, therefore, last-minute changes had to be made to be able to finish the project.

Another thing realized with the final product is, for the bucket cylinder it would be more suitable to use a linear actuator with a greater stroke, for example, 50 mm, as the others. Now the bucket can close properly but cannot open to its full potential. This does not have a big effect on the overall function, just a small detail that differs from a real excavator.

The problem with the excavator not being able to lift anything is mainly due to the way the rotational gear is constructed, and a little bit because of the lack of a counterweight. A counterweight is easy to place in the back of the excavator and a spot is prepared for it, but because of delays in delivery of parts there was no time left to find a counterweight. For the main problem, the gear, the problem is that it consists of two pieces that are not attached to each other in any way. They are placed on top of each other and stays in place if the center of gravity is positioned above it. When moving the arm, center of gravity changes and, therefore, the upper body start to tilt. A counterweight can work up to a certain limit, until it gets too heavy in the back when the bucket is unloaded causing the excavator to tilt backward instead. To solve this, the best option is to connect the two pieces, in a way that makes them constantly attached to each other. To do this, a part can be created to lock them together. This part would have to take all the forces created from the tilt of the upper body. See below for an example on how this part could look like.

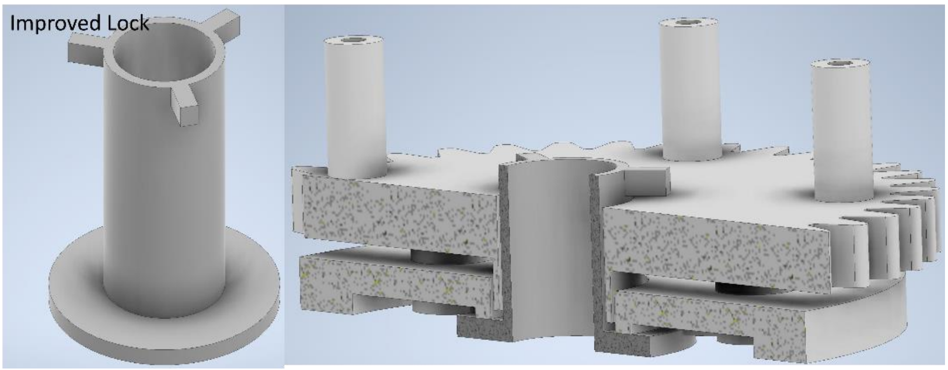


Figure 35: Improved rotational lock and assembled with gear

Another improvement that can be made is to upgrade the driving motors to some with a higher torque. In its current condition, the excavator can drive straight when both motors are working together, but individually they are too weak to move the excavator. The reason behind the use of the current motors is because they were already available at the university. Upgrading the motors would make turning the excavator easy, where you would be able to decide if only to turn with one side or use both tracks in opposite directions.

All things considered, the final product meets the requirements of the project and can be considered a success. A radio-controlled excavator can be built by using only 3D printing for the parts and adding the electronics.

7 Bibliography

[1] Lutkevich, B., 2019. *What is a Microcontroller and How Does it Work?*. [online] TechTarget. Available at: <https://www.techtarget.com/iotagenda/definition/microcontroller> [Accessed 4 May 2022].

[2] Docs.arduino.cc. n.d. *Arduino Mega 2560 Rev3*. [online] Available at: <https://docs.arduino.cc/hardware/mega-2560> [Accessed 4 May 2022].

[3] Arduino.cc. 2022. *Basics of PWM(Pulse Width Modulation)*. [online] Available at: <https://docs.arduino.cc/learn/microcontrollers/analog-output> [Accessed 4 May 2022].

[4] Arduino.cc. n.d. Software. [online] Available at: <https://www.arduino.cc/en/software> [Accessed 4 May 2022].

[5] Last Minute Engineers. n.d. In-Depth: Interface L298N DC Motor Driver Module with Arduino. [online] Available at: <https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/> [Accessed 4 May 2022].

[6] Ie.rs-online.com. n.d. Everything You Need To Know About DC Motors | RS. [online] Available at: <https://ie.rs-online.com/web/generalDisplay.html?id=ideas-and-advice/dc-motors-guide> [Accessed 5 May 2022].

[7] Dickson, R., 2018. Linear Actuators 101 - Everything You Need to Know About Linear Actuators. [online] Firgelli Automations. Available at: <https://www.firgelliauto.com/blogs/actuators/linear-actuators-101> [Accessed 5 May 2022].

[8] Arduino Project Hub. 2019. How to Interface PS2 Wireless Controller w/ Arduino. [online] Available at: <https://create.arduino.cc/projecthub/electropeak/how-to-interface-ps2-wireless-controller-w-arduino-a0a813?ref=search&ref_id=ps2%20controller&offset=1> [Accessed 5 May 2022].

[9] Schramme, C., 2020. Free CAD Designs, Files & 3D Models | The GrabCAD Community Library. [online] Grabcad.com. Available at: <https://grabcad.com/library/h-bridge-high-detail-1> [Accessed 1 June 2022].

[10] Engenharia, A., 2019. Free CAD Designs, Files & 3D Models | The GrabCAD Community Library. [online] Grabcad.com. Available at: <https://grabcad.com/library/arduino-mega-2560-r3-3> [Accessed 1 June 2022].

[11] Thingiverse.com. 2018. Rover Tracks v2 by nahueltaibo. [online] Available at: <https://www.thingiverse.com/thing:3112734/files> [Accessed 1 June 2022].

[12] Slipring, 2021. Carbon Brush Slip Ring. [Online] Available at: <https://slip-ring.com/what-is-an-electrical-slip-ring/> [Accessed 17 June 2022].

[13] Porter, B., 2010. PlayStation 2 Controller Arduino Library v1.0. [online] Billporter.info. Available at: <http://www.billporter.info/2010/06/05/playstation-2-controller-arduino-library-v1-0/> [Accessed 17 June 2022].

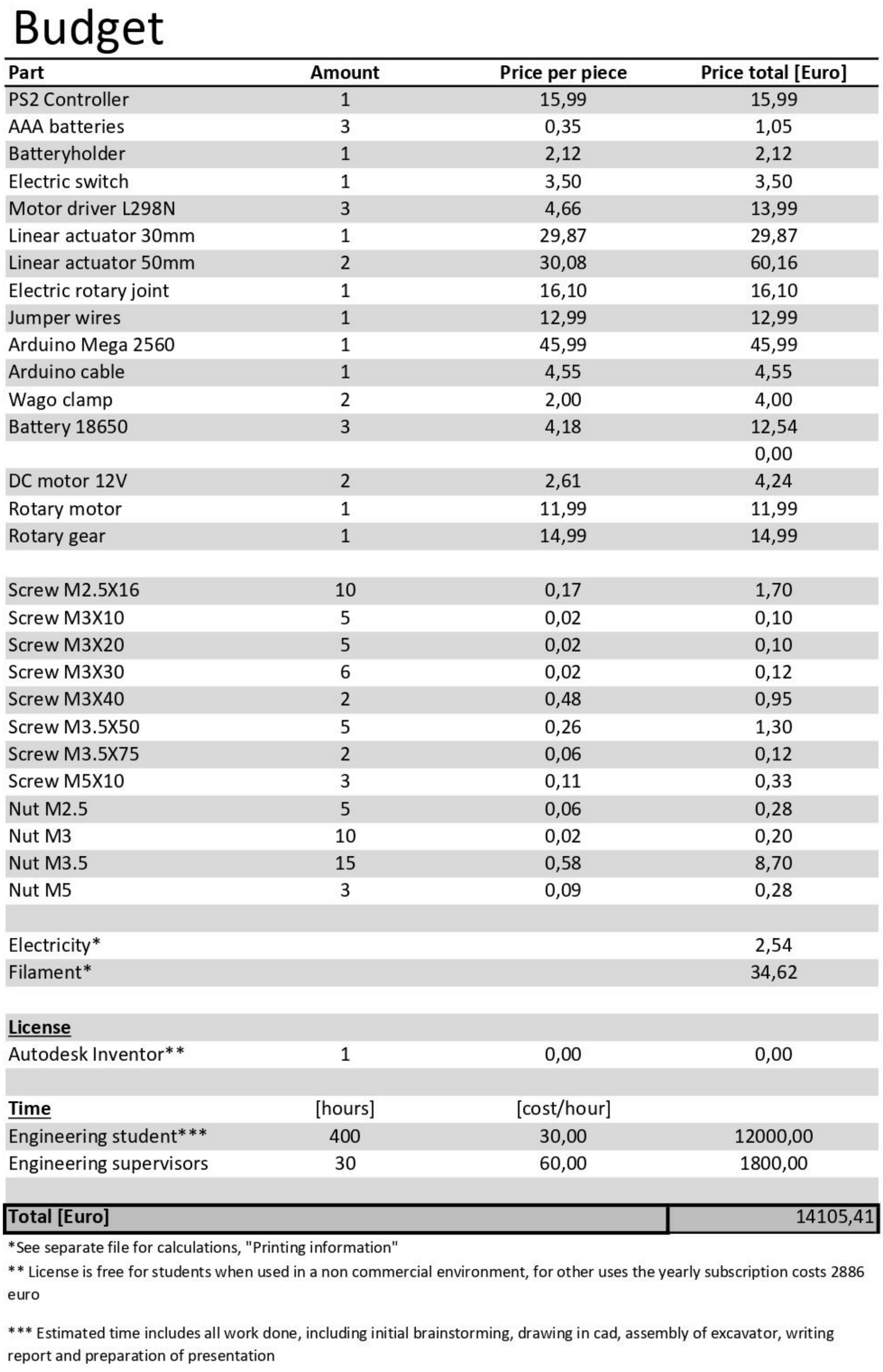
[14] n.d. Cirkit Designer. Cirkit studio. Available at: https://www.cirkitstudio.com/index.html

[15] Jenkins, B., n.d. DC Motors: Intro to Servos, BLDC motors, Steppers & More | Circuit. [online] Circuit Crush. Available at: <https://www.circuitcrush.com/intro-dc-motors/> [Accessed 3 July 2022].

[16] Ll-99, n.d. PS2 controller with description of buttons. [image] Available at: <https://ae01.alicdn.com/kf/H850b7e939bdb4f1d94f0f93474b107436/Baru-Nirkabel-Gamepad-untuk-Sony-PS2-Controller-untuk-Playstation-2-Konsol-Ganda-Joystick-Getaran-Shock-Joypad.jpg> [Accessed 5 July 2022].

8 Appendix

8.1 A. Budget



Budget

Part	Amount	Price per piece	Price total [Euro]
PS2 Controller	1	15,99	15,99
AAA batteries	3	0,35	1,05
Batteryholder	1	2,12	2,12
Electric switch	1	3,50	3,50
Motor driver L298N	3	4,66	13,99
Linear actuator 30mm	1	29,87	29,87
Linear actuator 50mm	2	30,08	60,16
Electric rotary joint	1	16,10	16,10
Jumper wires	1	12,99	12,99
Arduino Mega 2560	1	45,99	45,99
Arduino cable	1	4,55	4,55
Wago clamp	2	2,00	4,00
Battery 18650	3	4,18	12,54
			0,00
DC motor 12V	2	2,61	4,24
Rotary motor	1	11,99	11,99
Rotary gear	1	14,99	14,99
Screw M2.5X16	10	0,17	1,70
Screw M3X10	5	0,02	0,10
Screw M3X20	5	0,02	0,10
Screw M3X30	6	0,02	0,12
Screw M3X40	2	0,48	0,95
Screw M3.5X50	5	0,26	1,30
Screw M3.5X75	2	0,06	0,12
Screw M5X10	3	0,11	0,33
Nut M2.5	5	0,06	0,28
Nut M3	10	0,02	0,20
Nut M3.5	15	0,58	8,70
Nut M5	3	0,09	0,28
Electricity*			2,54
Filament*			34,62
License			
Autodesk Inventor**	1	0,00	0,00
Time	[hours]	[cost/hour]	
Engineering student***	400	30,00	12000,00
Engineering supervisors	30	60,00	1800,00
Total [Euro]			14105,41

*See separate file for calculations, "Printing information"

** License is free for students when used in a non commercial environment, for other uses the yearly subscription costs 2886 euro

*** Estimated time includes all work done, including initial brainstorming, drawing in cad, assembly of excavator, writing report and preparation of presentation

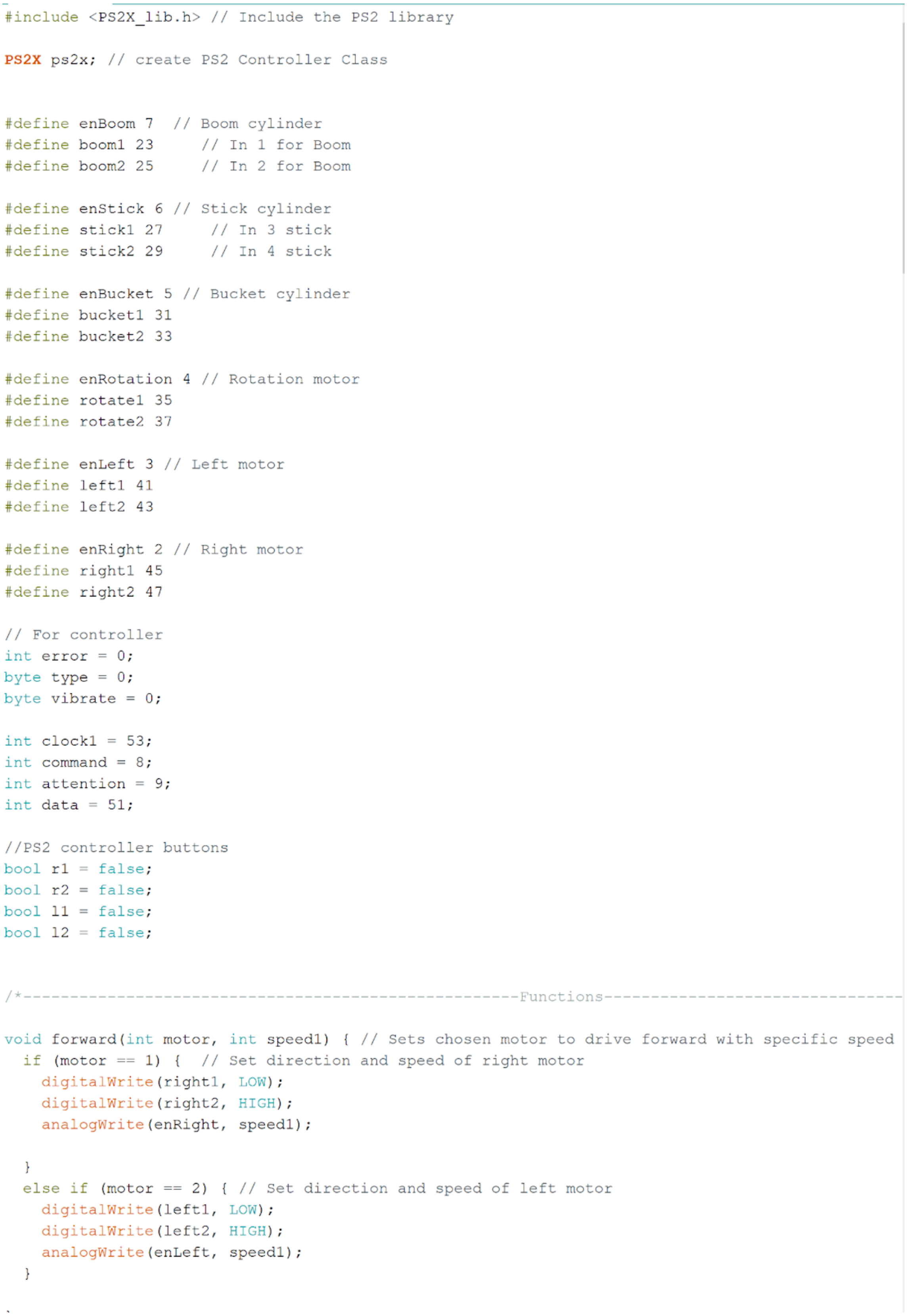
Printing information

Part	Time [min]	Material length [m]	Material weight [g]
Arm			
Boom x2	221	21,05	62,85
Stick	190	12,73	37,96
Bucket	189	10,44	31,31
Tipping link	8	0,415	1,2
Bucket link	8	0,415	1,2
Arm attachment	199	15	44,7
Cabin			
Cabin body + roof	450	27	80
Cabin chair	212	4,7	14,18
Cabin display	57	0,84	2,48
Cabin door	77	1,5	4,43
Tracks			
Distance x8	10	0,4	1,23
Distance lock x4	14	0,9	2,62
Gears x8	152	11,12	32,84
Lock x2	214	19,28	57,58
Side chassi x2	416	31,44	92,7
Side inner x2	400	22,98	68,6
Side outer x2	344	25,36	75,66
Links x76	1160	51,325	153,2
Body			
Base plate	420	35,5	106,27
Body plate	361	32,91	97,955
Cover	650	48,7	145,25
Engine cover	156	11,94	35,67
Exhaust	21	1	2,6
Rotational lock	23	0,75	2,05
TOTAL	5952	387,695	1154,535
	99h 12 min		1,15 kg

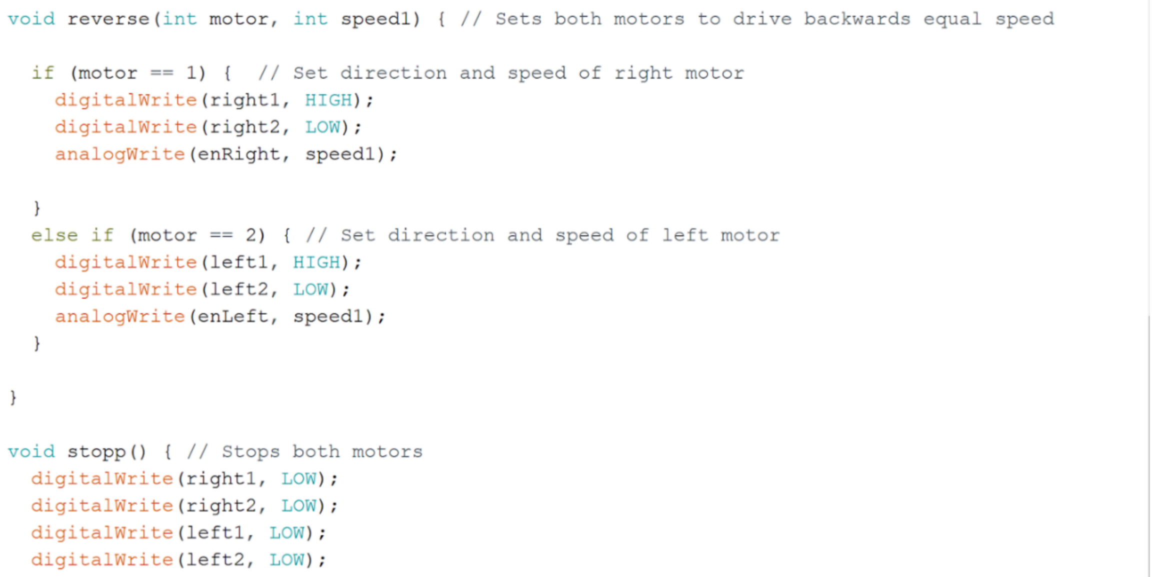
Price of electricity:	0,32	euro / kWh
Price of filament (1kg)	29,99	euro
Power consumption, prusa printer	80W	

Cost of energy:	2,54	Euro
Cost of filament:	34,62	Euro

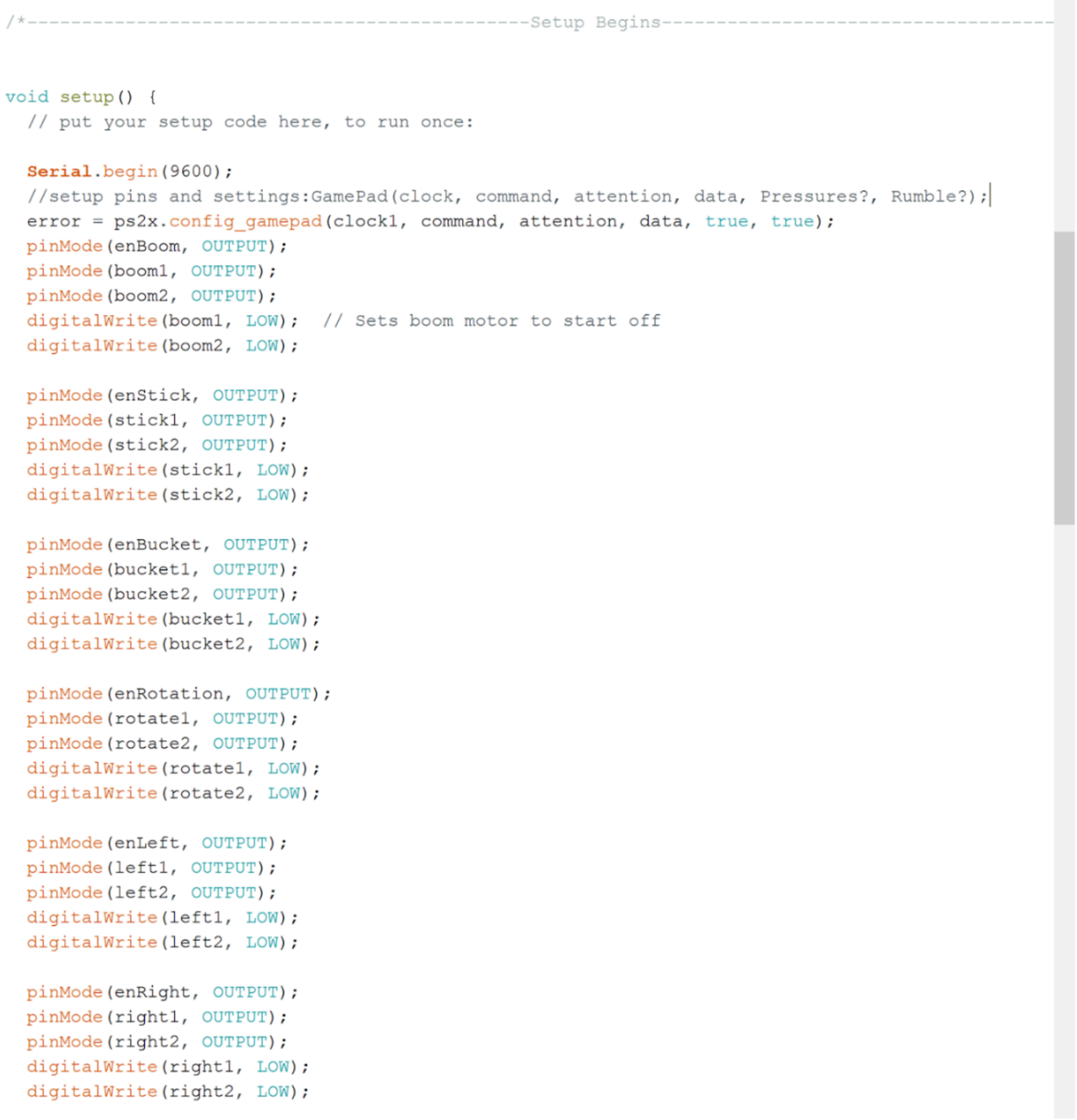
8.2 B. Arduino Code



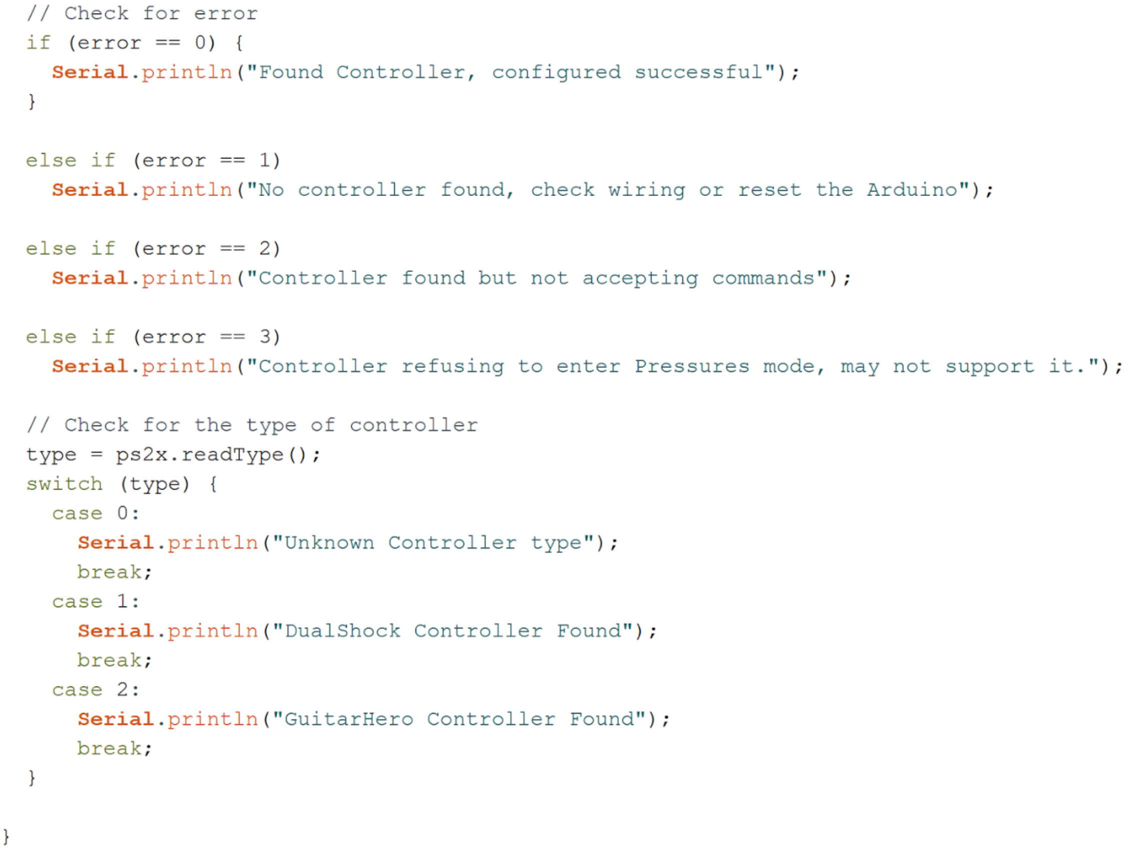
```
#include <PS2X_lib.h> // Include the PS2 library

PS2X ps2x; // create PS2 Controller Class


#define enBoom 7  // Boom cylinder
#define boom1 23     // In 1 for Boom
#define boom2 25     // In 2 for Boom

#define enStick 6 // Stick cylinder
#define stick1 27     // In 3 stick
#define stick2 29     // In 4 stick

#define enBucket 5 // Bucket cylinder
#define bucket1 31
#define bucket2 33

#define enRotation 4 // Rotation motor
#define rotate1 35
#define rotate2 37

#define enLeft 3 // Left motor
#define left1 41
#define left2 43

#define enRight 2 // Right motor
#define right1 45
#define right2 47

// For controller
int error = 0;
byte type = 0;
byte vibrate = 0;

int clock1 = 53;
int command = 8;
int attention = 9;
int data = 51;

//PS2 controller buttons
bool r1 = false;
bool r2 = false;
bool l1 = false;
bool l2 = false;


/*-----------------------------------------------------Functions--------------------------------

void forward(int motor, int speed1) { // Sets chosen motor to drive forward with specific speed
  if (motor == 1) {  // Set direction and speed of right motor
    digitalWrite(right1, LOW);
    digitalWrite(right2, HIGH);
    analogWrite(enRight, speed1);

  }
  else if (motor == 2) { // Set direction and speed of left motor
    digitalWrite(left1, LOW);
    digitalWrite(left2, HIGH);
    analogWrite(enLeft, speed1);
  }
```



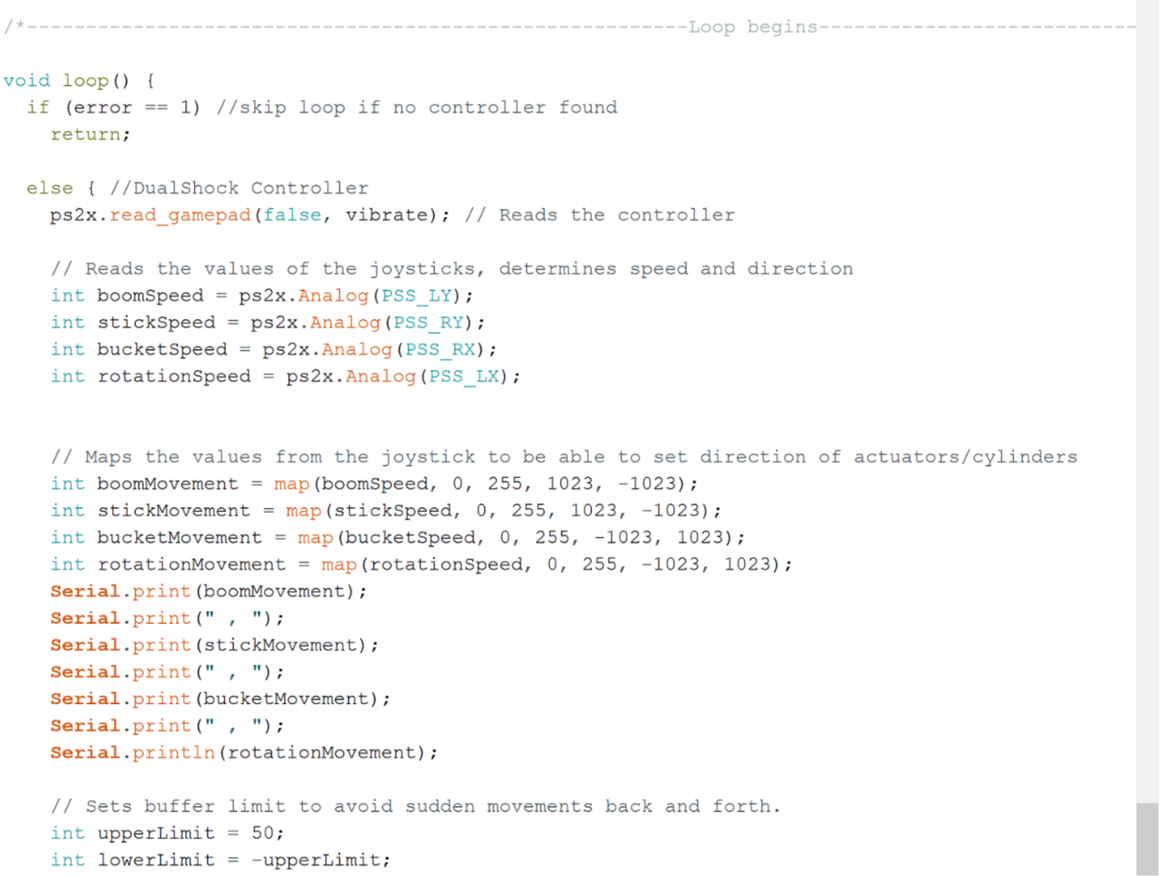
```
void reverse(int motor, int speed1) { // Sets both motors to drive backwards equal speed

  if (motor == 1) {  // Set direction and speed of right motor
    digitalWrite(right1, HIGH);
    digitalWrite(right2, LOW);
    analogWrite(enRight, speed1);

  }
  else if (motor == 2) { // Set direction and speed of left motor
    digitalWrite(left1, HIGH);
    digitalWrite(left2, LOW);
    analogWrite(enLeft, speed1);
  }

}

void stopp() { // Stops both motors
  digitalWrite(right1, LOW);
  digitalWrite(right2, LOW);
  digitalWrite(left1, LOW);
  digitalWrite(left2, LOW);
```



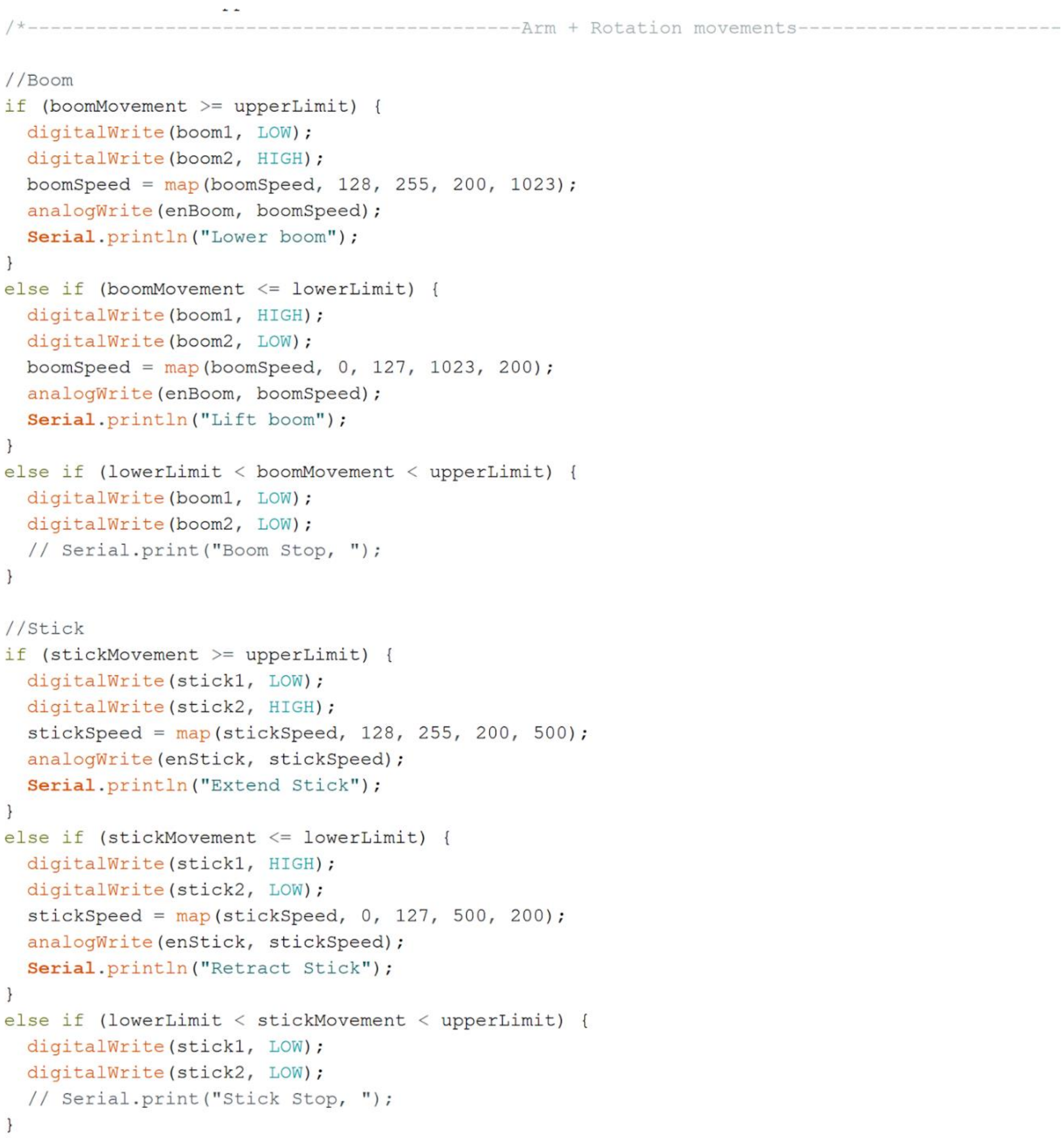
```
/*----------------------------------------------Setup Begins-----------------------------------


void setup() {
  // put your setup code here, to run once:

  Serial.begin(9600);
  //setup pins and settings:GamePad(clock, command, attention, data, Pressures?, Rumble?);
  error = ps2x.config_gamepad(clock1, command, attention, data, true, true);
  pinMode(enBoom, OUTPUT);
  pinMode(boom1, OUTPUT);
  pinMode(boom2, OUTPUT);
  digitalWrite(boom1, LOW);  // Sets boom motor to start off
  digitalWrite(boom2, LOW);

  pinMode(enStick, OUTPUT);
  pinMode(stick1, OUTPUT);
  pinMode(stick2, OUTPUT);
  digitalWrite(stick1, LOW);
  digitalWrite(stick2, LOW);

  pinMode(enBucket, OUTPUT);
  pinMode(bucket1, OUTPUT);
  pinMode(bucket2, OUTPUT);
  digitalWrite(bucket1, LOW);
  digitalWrite(bucket2, LOW);

  pinMode(enRotation, OUTPUT);
  pinMode(rotate1, OUTPUT);
  pinMode(rotate2, OUTPUT);
  digitalWrite(rotate1, LOW);
  digitalWrite(rotate2, LOW);

  pinMode(enLeft, OUTPUT);
  pinMode(left1, OUTPUT);
  pinMode(left2, OUTPUT);
  digitalWrite(left1, LOW);
  digitalWrite(left2, LOW);

  pinMode(enRight, OUTPUT);
  pinMode(right1, OUTPUT);
  pinMode(right2, OUTPUT);
  digitalWrite(right1, LOW);
  digitalWrite(right2, LOW);
```



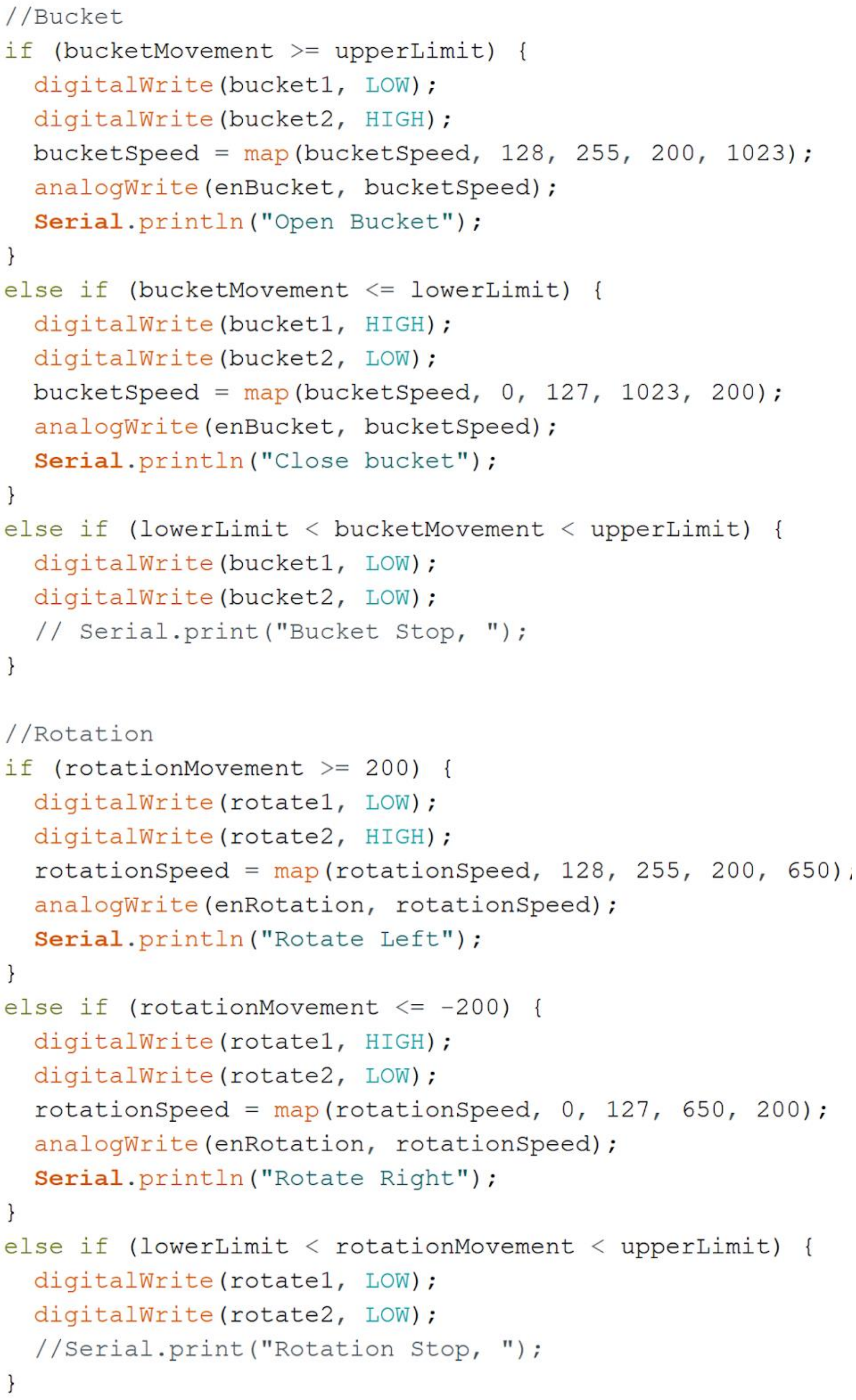
```
  // Check for error
  if (error == 0) {
    Serial.println("Found Controller, configured successful");
  }

  else if (error == 1)
    Serial.println("No controller found, check wiring or reset the Arduino");

  else if (error == 2)
    Serial.println("Controller found but not accepting commands");

  else if (error == 3)
    Serial.println("Controller refusing to enter Pressures mode, may not support it.");

  // Check for the type of controller
  type = ps2x.readType();
  switch (type) {
    case 0:
      Serial.println("Unknown Controller type");
      break;
    case 1:
      Serial.println("DualShock Controller Found");
      break;
    case 2:
      Serial.println("GuitarHero Controller Found");
      break;
  }

}
```



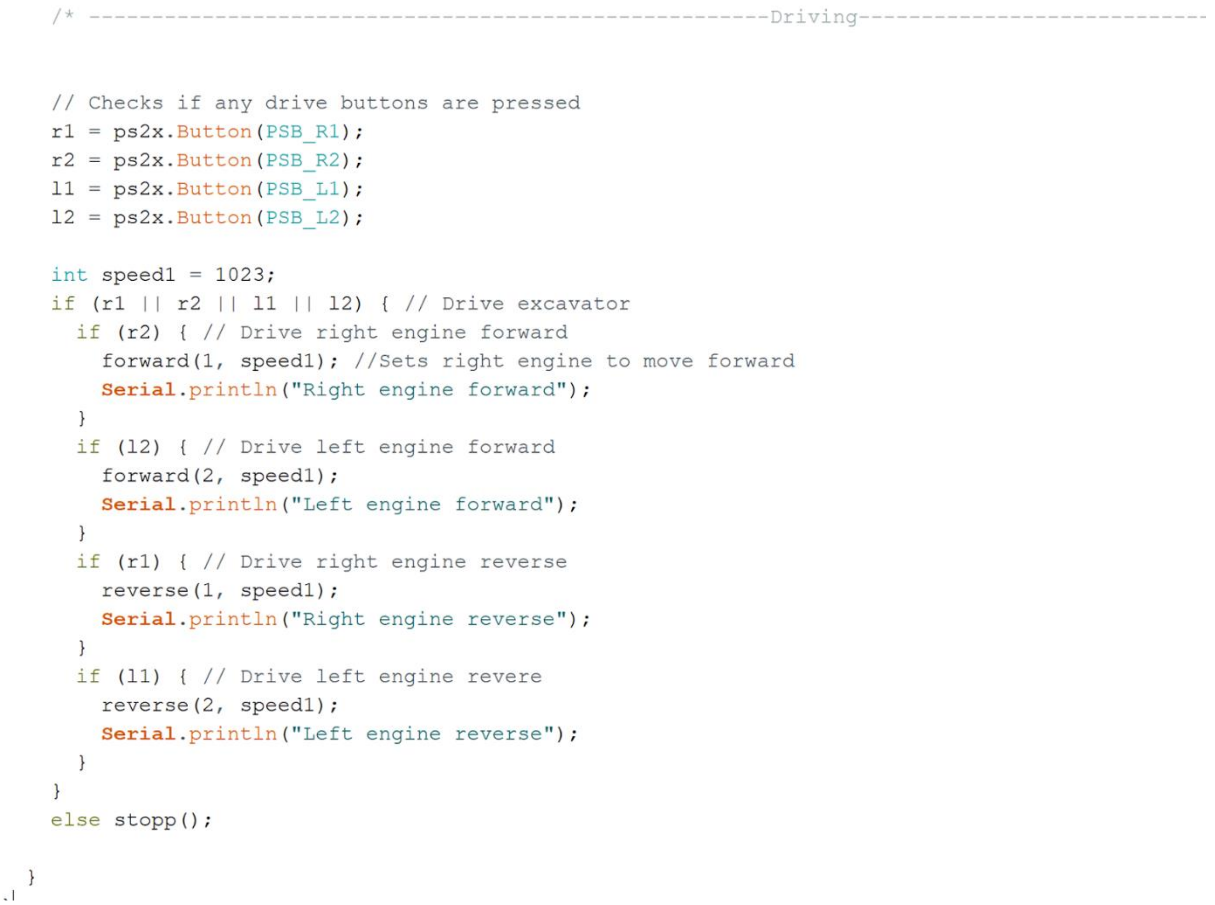
```
/*--------------------------------------------------------Loop begins---------------------------

void loop() {
  if (error == 1) //skip loop if no controller found
    return;

  else { //DualShock Controller
    ps2x.read_gamepad(false, vibrate); // Reads the controller

    // Reads the values of the joysticks, determines speed and direction
    int boomSpeed = ps2x.Analog(PSS_LY);
    int stickSpeed = ps2x.Analog(PSS_RY);
    int bucketSpeed = ps2x.Analog(PSS_RX);
    int rotationSpeed = ps2x.Analog(PSS_LX);


    // Maps the values from the joystick to be able to set direction of actuators/cylinders
    int boomMovement = map(boomSpeed, 0, 255, 1023, -1023);
    int stickMovement = map(stickSpeed, 0, 255, 1023, -1023);
    int bucketMovement = map(bucketSpeed, 0, 255, -1023, 1023);
    int rotationMovement = map(rotationSpeed, 0, 255, -1023, 1023);
    Serial.print(boomMovement);
    Serial.print(" , ");
    Serial.print(stickMovement);
    Serial.print(" , ");
    Serial.print(bucketMovement);
    Serial.print(" , ");
    Serial.println(rotationMovement);

    // Sets buffer limit to avoid sudden movements back and forth.
    int upperLimit = 50;
    int lowerLimit = -upperLimit;
```



```
/*-------------------------------------------Arm + Rotation movements-----------------------

//Boom
if (boomMovement >= upperLimit) {
  digitalWrite(boom1, LOW);
  digitalWrite(boom2, HIGH);
  boomSpeed = map(boomSpeed, 128, 255, 200, 1023);
  analogWrite(enBoom, boomSpeed);
  Serial.println("Lower boom");
}
else if (boomMovement <= lowerLimit) {
  digitalWrite(boom1, HIGH);
  digitalWrite(boom2, LOW);
  boomSpeed = map(boomSpeed, 0, 127, 1023, 200);
  analogWrite(enBoom, boomSpeed);
  Serial.println("Lift boom");
}
else if (lowerLimit < boomMovement < upperLimit) {
  digitalWrite(boom1, LOW);
  digitalWrite(boom2, LOW);
  // Serial.print("Boom Stop, ");
}

//Stick
if (stickMovement >= upperLimit) {
  digitalWrite(stick1, LOW);
  digitalWrite(stick2, HIGH);
  stickSpeed = map(stickSpeed, 128, 255, 200, 500);
  analogWrite(enStick, stickSpeed);
  Serial.println("Extend Stick");
}
else if (stickMovement <= lowerLimit) {
  digitalWrite(stick1, HIGH);
  digitalWrite(stick2, LOW);
  stickSpeed = map(stickSpeed, 0, 127, 500, 200);
  analogWrite(enStick, stickSpeed);
  Serial.println("Retract Stick");
}
else if (lowerLimit < stickMovement < upperLimit) {
  digitalWrite(stick1, LOW);
  digitalWrite(stick2, LOW);
  // Serial.print("Stick Stop, ");
}
```

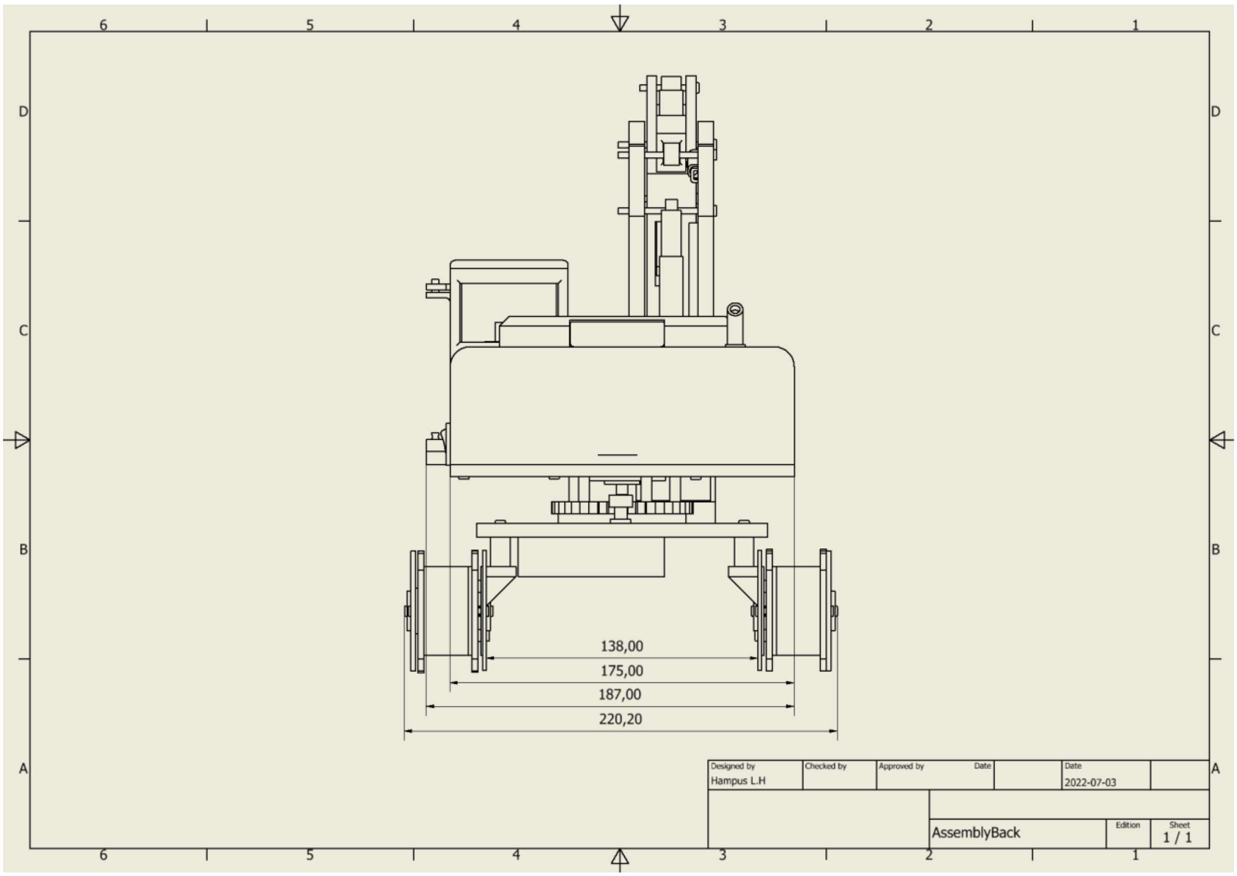


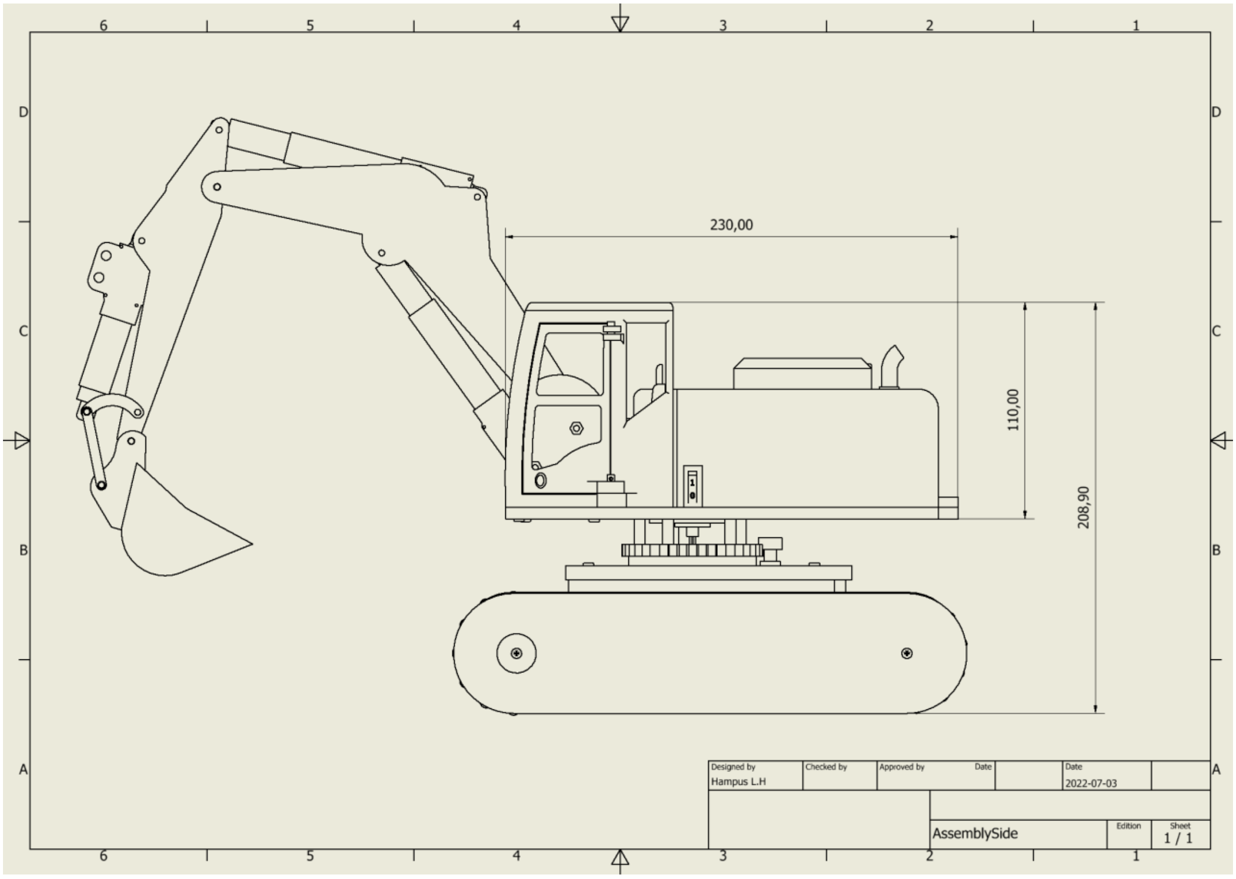
```
//Bucket
if (bucketMovement >= upperLimit) {
  digitalWrite(bucket1, LOW);
  digitalWrite(bucket2, HIGH);
  bucketSpeed = map(bucketSpeed, 128, 255, 200, 1023);
  analogWrite(enBucket, bucketSpeed);
  Serial.println("Open Bucket");
}
else if (bucketMovement <= lowerLimit) {
  digitalWrite(bucket1, HIGH);
  digitalWrite(bucket2, LOW);
  bucketSpeed = map(bucketSpeed, 0, 127, 1023, 200);
  analogWrite(enBucket, bucketSpeed);
  Serial.println("Close bucket");
}
else if (lowerLimit < bucketMovement < upperLimit) {
  digitalWrite(bucket1, LOW);
  digitalWrite(bucket2, LOW);
  // Serial.print("Bucket Stop, ");
}

//Rotation
if (rotationMovement >= 200) {
  digitalWrite(rotate1, LOW);
  digitalWrite(rotate2, HIGH);
  rotationSpeed = map(rotationSpeed, 128, 255, 200, 650);
  analogWrite(enRotation, rotationSpeed);
  Serial.println("Rotate Left");
}
else if (rotationMovement <= -200) {
  digitalWrite(rotate1, HIGH);
  digitalWrite(rotate2, LOW);
  rotationSpeed = map(rotationSpeed, 0, 127, 650, 200);
  analogWrite(enRotation, rotationSpeed);
  Serial.println("Rotate Right");
}
else if (lowerLimit < rotationMovement < upperLimit) {
  digitalWrite(rotate1, LOW);
  digitalWrite(rotate2, LOW);
  //Serial.print("Rotation Stop, ");
}
```

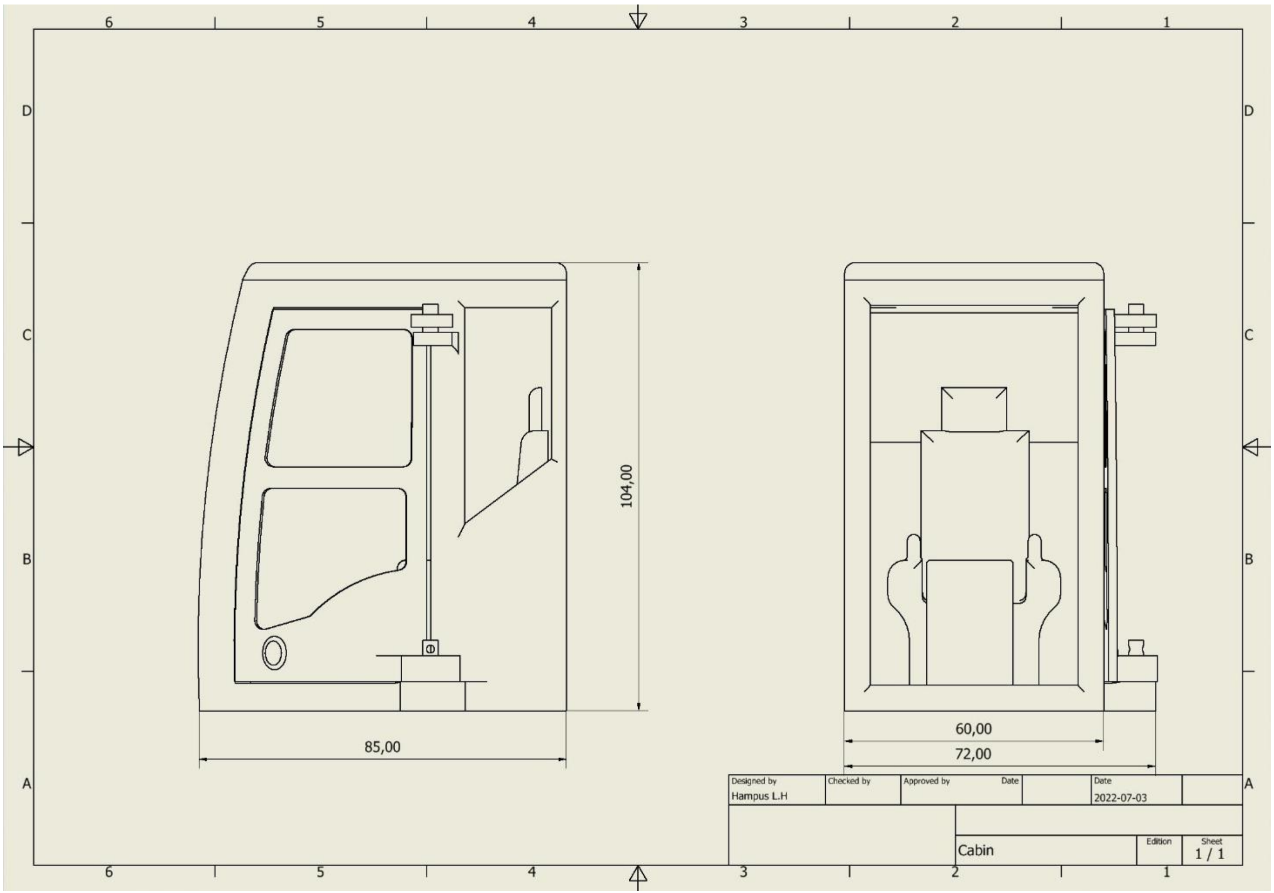


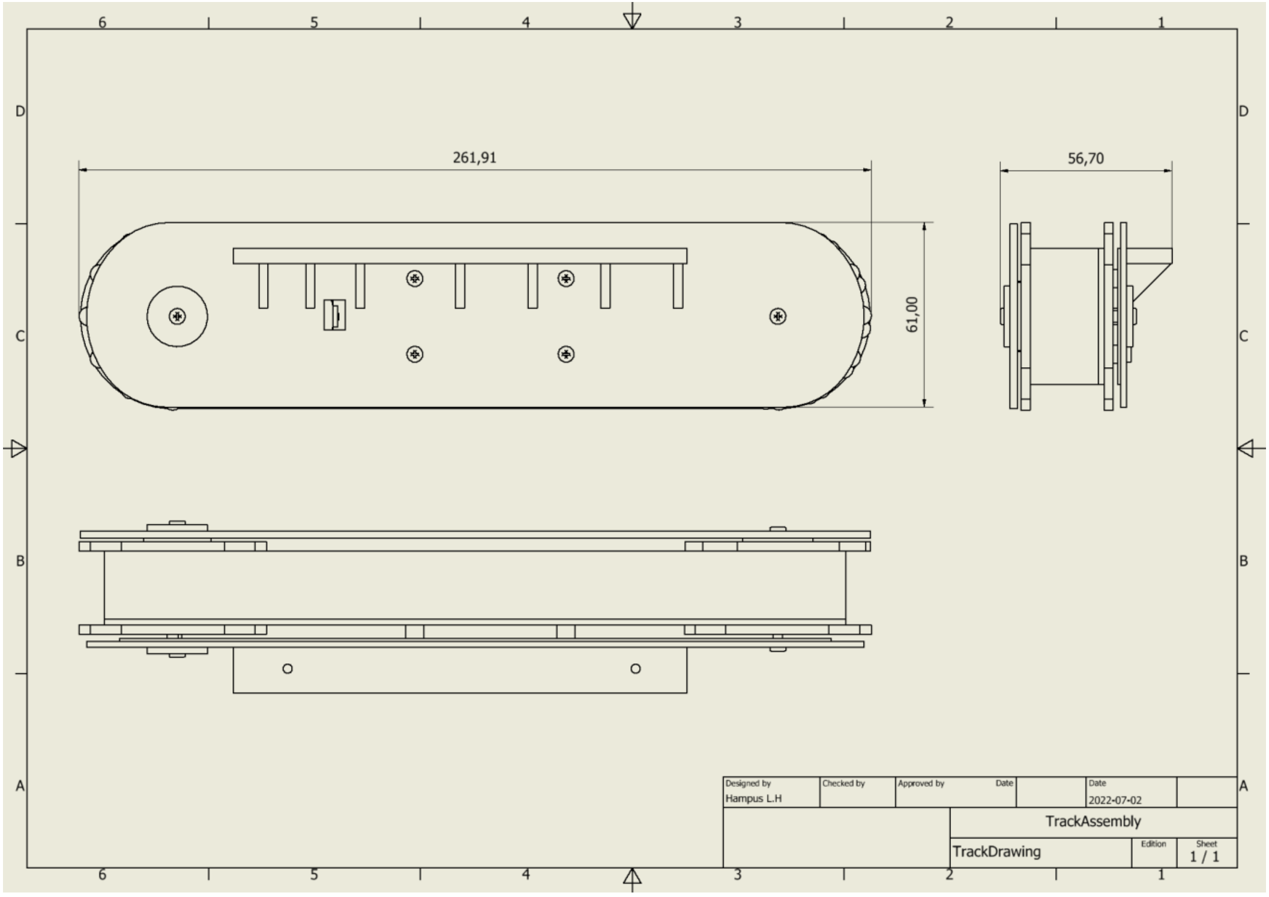
```
  /* ------------------------------------------------------Driving----------------------------

  // Checks if any drive buttons are pressed
  r1 = ps2x.Button(PSB_R1);
  r2 = ps2x.Button(PSB_R2);
  l1 = ps2x.Button(PSB_L1);
  l2 = ps2x.Button(PSB_L2);

  int speed1 = 1023;
  if (r1 || r2 || l1 || l2) { // Drive excavator
    if (r2) { // Drive right engine forward
      forward(1, speed1); //Sets right engine to move forward
      Serial.println("Right engine forward");
    }
    if (l2) { // Drive left engine forward
      forward(2, speed1);
      Serial.println("Left engine forward");
    }
    if (r1) { // Drive right engine reverse
      reverse(1, speed1);
      Serial.println("Right engine reverse");
    }
    if (l1) { // Drive left engine revere
      reverse(2, speed1);
      Serial.println("Left engine reverse");
    }
  }
  else stopp();

}
```

8.3 C. Dimensions

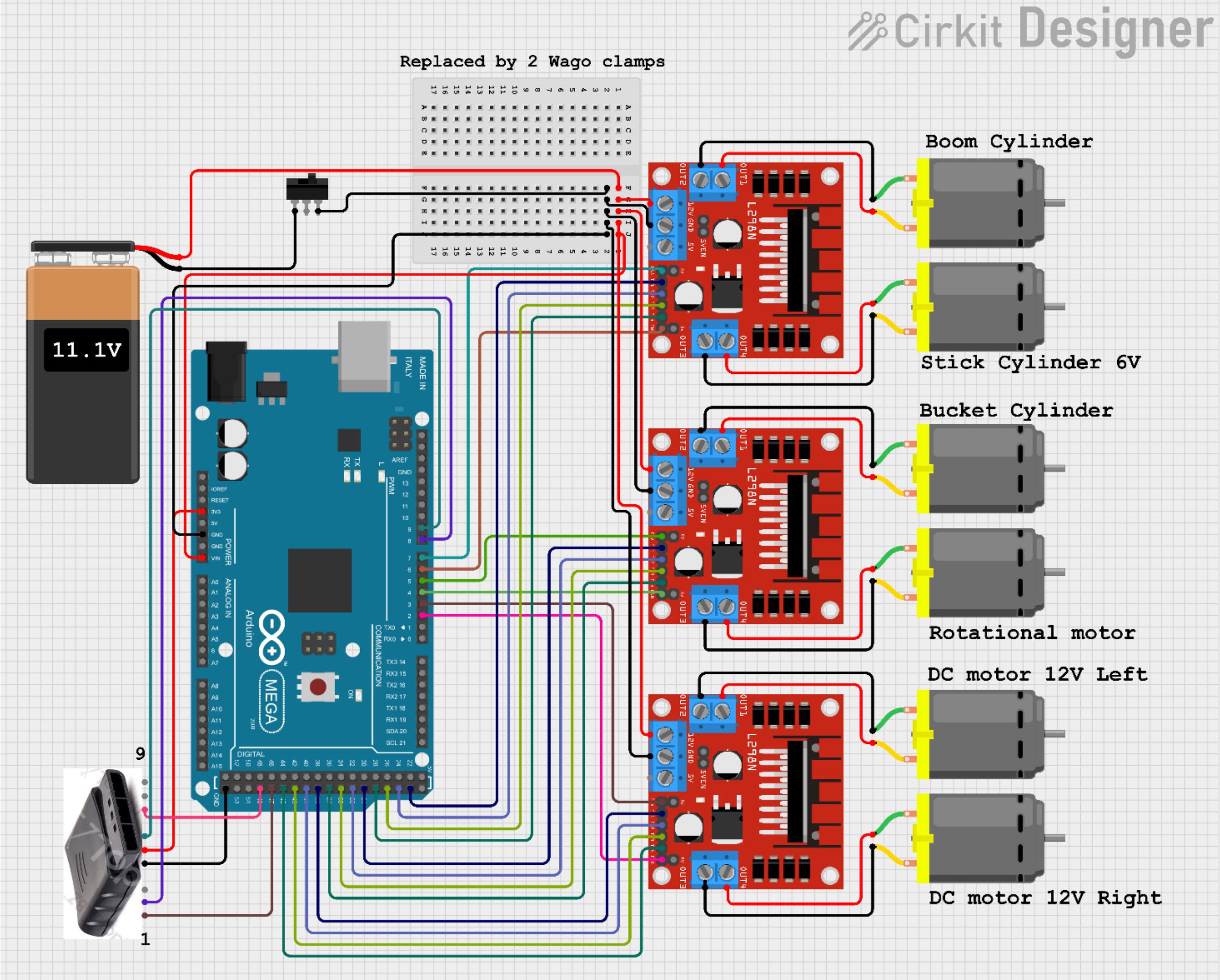





6
5
4
3
2
1
D
C
B
A
104,00
85,00
60,00
72,00
Designed by
Hampus L.H
Checked by
Approved by
Date
Date
2022-07-03
Cabin
Edition
Sheet
1 / 1


6
5
4
3
2
1
D
C
B
A
261,91
56,70
61,00
Designed by
Hampus L.H
Checked by
Approved by
Date
Date
2022-07-02
TrackAssembly
TrackDrawing
Edition
Sheet
1 / 1

8.4 D. Electrical Circuit



Made using Cirkit Designer [14]

8.5 E. Videos

Playlist of operational videos: YouTube Playlist of Operational Videos

Exploded view of track: https://youtu.be/NxzTpPXTUhc

8.6 F. Arduino MEGA technical specification

MICROCONTROLLER	ATmega2560
OPERATING VOLTAGE	5V
INPUT VOLTAGE (RECOMMENDED)	7-12V
INPUT VOLTAGE (LIMIT)	6-20V
DIGITAL I/O PINS	54 (of which 15 provide PWM output)
ANALOG INPUT PINS	16
DC CURRENT PER I/O PIN	20 mA
DC CURRENT FOR 3.3V PIN	50 mA
FLASH MEMORY	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
CLOCK SPEED	16 MHz
LED_BUILTIN	13
LENGTH	101.52 mm
WIDTH	53.3 mm
WEIGHT	37 g

(Source: https://store.arduino.cc/products/arduino-mega-2560-rev3)