AUTOMATED NEONATAL EXCHANGE TRANSFUSION

Prince Albert Kwasi Okrah

Benjamin Amanor Kisser

Department of Computer Engineering

College of Engineering

Kwame Nkrumah University of Science and Technology

April 26, 2021

Supervised by: Mr. Prince Odame

DECLARATION

We hereby declare that this project is our own original work towards the partial fulfilment of the requirement for the award of an undergraduate Bachelor of Science degree (BSc.) in Biomedical Engineering. We also declare that, to the very best of our knowledge, this report contains no material that has been previously published by another person, neither does it contain any material which has been accepted for the award of any other degree of the university, except where due acknowledgment has been made in the text

Prince Albert Kwasi Okrah ………………………….. 17/09/2021

Benjamin Amanor Kisser ………………………..… ……. 17/09/2021

Certified By:

Mr. Prince Odame ………………………………. ……………….

(Supervisor)

ACKNOWLEDGEMENT

To God be the glory for the great things He has done. We are most grateful to God for endowing us with the wisdom and understanding to undertake this project. We appreciate our project supervisor, Mr. Prince Odame, for his continuous encouragement, guidance, and support throughout the project. We say a big “thank you” to our parents for their prayers, advice, and contributions to the project. Finally, we would like to thank Godwin Wunpiini Yiddi and Emmanuel Amankwah for setting the pace for us to follow. Your excellent work is commendable. May the good Lord continue to bless you all.

DEDICATION

This project is dedicated to all medical physicians who, despite the complications associated with exchange transfusion, continue to help babies with severe hyperbilirubinemia. You are the reason many families smile today.

ABSTRACT

Blood disorders such as hyperbilirubinemia have long been the cause of death for many babies in Ghana and in most parts of Africa, where there is a deficit in the sophisticated equipment necessary for the early detection and elimination of such disorders. For mild hyperbilirubinemia conditions, phototherapy and intravenous immunoglobin transfusion (IVIG) catered for them and saved many lives.

Nevertheless, IVIG and phototherapy could not deal with all life-threatening blood abnormalities; exchange transfusion (ET) was the go-to solution. It should be noted, however, that exchange transfusion is a time-consuming and stressful procedure that lasts approximately two hours. This process leaves many physicians not wanting to perform the complicated procedure.

This project, therefore, sought to combine mechanics, electronics, and programming to design and build a low-cost device that automates the neonatal exchange transfusion procedure to eliminate human errors, lessen the burden on physicians, and, most importantly, increase the chances of survival for babies with hyperbilirubinemia. The approach involved integrating the existing manual ET process with the proposed design. A comparison between the automated neonate exchange transfusion (ANET) device and the manual process confirmed the level of consistency the device provides throughout the exchange transfusion process. Recommendations have been made for further work on the project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

The introductory chapter talks about the background to the project, the relevance, scope and objectives, and its significance to the advancement of the health sector in Ghana.

1.1 BACKGROUND

Hyperbilirubinemia in neonates is a leading cause of lifelong neurodevelopmental impairment (Alkén J, 2019). Statistics show that over 50% of term infants and 80% of preterm infants develop jaundice, usually between two and four days after birth (Luke A. Jardine, 2011). Most new-borns experience this because they have very high levels of serum bilirubin (SBR) during their first week of life (L M Gartner, 1977). The high levels of bilirubin in new-borns are as a result of immature internal body organs like the liver which is responsible for reducing the levels of bilirubin in the blood. Luke A. Jardine (2011) further reveals that this condition however does not usually prevail for long as it may disappear on its own. However, this is not the case for some babies. To counter the effects of hyperbilirubinemia, a number of therapeutic modalities have evolved throughout the years. The most predominant one is exchange transfusion. Phototherapy becomes less effective for treating babies with severe hyperbilirubinemia (serum bilirubin levels exceeding 20mg/dL) (Brian Berman, 1979). Research shows that exchange transfusion in the western world is seldom performed due to sophisticated interventions such as the use of phototherapy (use of controlled light sources to treat certain medical conditions) and intravenous immunoglobin transfusion. Nevertheless, the effects of hyperbilirubinemia still occur in these highly-resourced countries in spite of evidence-based guidelines for treatment and prevention. On the other hand, most hospitals in Ghana are under resourced in terms of modern hospital equipment. Treatment of jaundice is primarily done by phototherapy and exchange transfusion.

Current data suggests that no public hospital in Ghana has a highly intense phototherapy unit; the kind that can treat severe hyperbilirubinemia. Exchange transfusion has therefore been the last resort for babies with hyperbilirubinemia in our part of the world. For this reason, there is a need to look into this life-saving procedure to ensure its efficiency and efficacy.

1.2 PROBLEM STATEMENT

In Ghana and other parts of Africa and the world at large, neonatal exchange transfusion is usually carried out by a physician and an assisting nurse(s). The procedure takes approximately two hours to perform, during which the physician will have to draw out and pump blood into the baby's body. (National Library Of Medicines, 2014). This must be done very slowly and steadily in cycles to avoid iron overload, which can cause damage to the heart, liver and other organs in the baby's body. The slow and long process makes the procedure tedious and most often overwhelming. Also, the procedure is susceptible to human errors as there is no guarantee that the same aliquots of blood will be delivered at the same steady rate throughout the two-hour long process. In some cases, doctors may not even want to perform it. Meanwhile, it is an effective life-saving procedure for babies with life-threatening blood abnormalities. There is therefore the need to automate the procedure. *Figure 1* below shows a baby undergoing manual exchange transfusion.

Figure 1:Neonate undergoing exchange transfusion

1.3 OBJECTIVES

The key objective of this project is to design and build a device that automates the neonatal exchange transfusion process.

SPECIFIC OBJECTIVES

The specific objectives of this project are to:

I. Combine mechanics, electronics and programming to control the entire exchange transfusion set-up with little or no human interference.
II. Incorporate all the details used in the manual process into our design
III. Program an informative interface
IV. Produce a low-cost and energy-saving device

1.4 SCOPE OF THE PROJECT

The movement of the syringe to draw blood and deliver blood will be controlled by a motor at a steady rate using programming. The manual switching of valves by the physician during the exchange procedure will also be replaced by motors. This will almost eliminate any human intervention in the procedure.

1.5 SIGNIFICANCE OF THE PROJECT

The significance of this project cannot be overemphasized. A less human-dependent device means a higher guarantee for consistency, accuracy and efficiency. Also, the automation of the neonatal exchange transfusion process will bring relief to many doctors and health workers. Furthermore, its cost as compared to many other medical devices is relatively low, and can be purchased by the average hospital in Ghana.

CHAPTER TWO

LITERATURE REVIEW

This chapter details on exchange transfusion – history, procedure, techniques and likely complications associated with the procedure.

2.1 EXCHANGE TRANSFUSION

Exchange blood transfusion is a medical procedure carried out by removing and replacing blood from one's body with blood or plasma from a donor (Kaneshiro, 2021). In most cases it involves inserting one or more catheters into a blood vessel. It must be noted that the exchange transfusion procedure is done in cycles where the baby's blood is slowly withdrawn (often about 5 to 20 ml at a time, depending on the baby's size and the severity of the illness) (Corinna Underwood, 2019). An equal amount of fresh, prewarmed blood or plasma flows into the baby's body. The cycle is repeated until the correct volume of blood has been successfully replaced. It is a life-saving procedure executed to counteract the effects of life-threatening blood abnormalities. Some of these life-threatening blood abnormalities which include jaundice, severe malaria, haemolytic disease of new-borns among others.

2.2 HISTORY OF EXCHANGE TRANSFUSION

Blood exchange transfusion (BET) was introduced in the late 1920s. The procedure was targeted at decreasing the mortality attributable to rhesus haemolytic disease of new-borns and to prevent kernicterus (a rare type of brain damage that occurs in a newborn with severe jaundice) in surviving infants (Teichler-Zallen D, 2004). A physician from Toronto's Hospital for Sick Children, Dr. Alfred P. Hart, in 1925 was the first to perform an exchange transfusion to treat haemolytic disease of the new-born (Elizabeth, 1975). The technique, exchange transfusion, was originally developed and named by Alexander Weiner, even though it was hinted by a Canadian surgeon, Dr. Bruce Robertson in 1921. The procedure was used in the treatment of severe burns in children. It involved the withdrawal of blood from the femoral vein, external jugular vein, longitudinal sinus, or radial artery of the young child and replacing it with fresh donor blood by way of a cut-down at the ankle. This procedure and a few other modifications of it were technically difficult, had significant complications and went unnoticed and unused, until 1946. Later, the procedure was supplanted by a method introduced by Dr Louis K. Diamond. In his method, he used the umbilical vein only to perform the exchange transfusion (Cogan, 1978). According to LK (1947), this procedure was relatively simpler and safer compared to the earlier technique and was subsequently adopted around the world. It made an enormous historical influence in neonatology. Exchange transfusion became so much common. In later years, the introduction of phototherapy and other medications have reduced the need for exchange transfusion. When phototherapy is initiated early, it is usually effective at controlling hyperbilirubinemia in infants (LK, 1947). The use of antenatal anti-Rh D antibody prophylaxis and screening for iso-sensitization has reduced the occurrence of haemolytic disease of the new-born, hence a reduction in the need for an exchange transfusion (M Funato, 1997).

2.3 RELEVANCE

While exchange transfusion is performed for both children and adults, it is most commonly done in children especially neonates. The process is called neonatal exchange transfusion. When other therapeutic modalities such as early and intensive use of phototherapy prove to be ineffective, a physician, with the consent of the parent of the baby, performs exchange transfusion. This procedure is usually the last resort for the treatment of haemolytic diseases of new-borns and hyperbilirubinemia.

The exchange transfusion procedure is performed for a number of reasons based on the problem being addressed. In the case of hyperbilirubinemia, it is to reduce the serum bilirubin (SBR) level and further lower the risk of brain damage. For haemolytic disease of the new-born, exchange transfusion is performed to remove the affected red blood cells of the infant and the circulating maternal antibodies to reduce destruction of the baby’s red blood cells. Physicians also use the procedure for treating potential heart failure while maintaining a normal amount of blood and also to correct anaemia (M Funato, 1997). *Figure 2* below shows the difference between healthy and jaundiced babies.

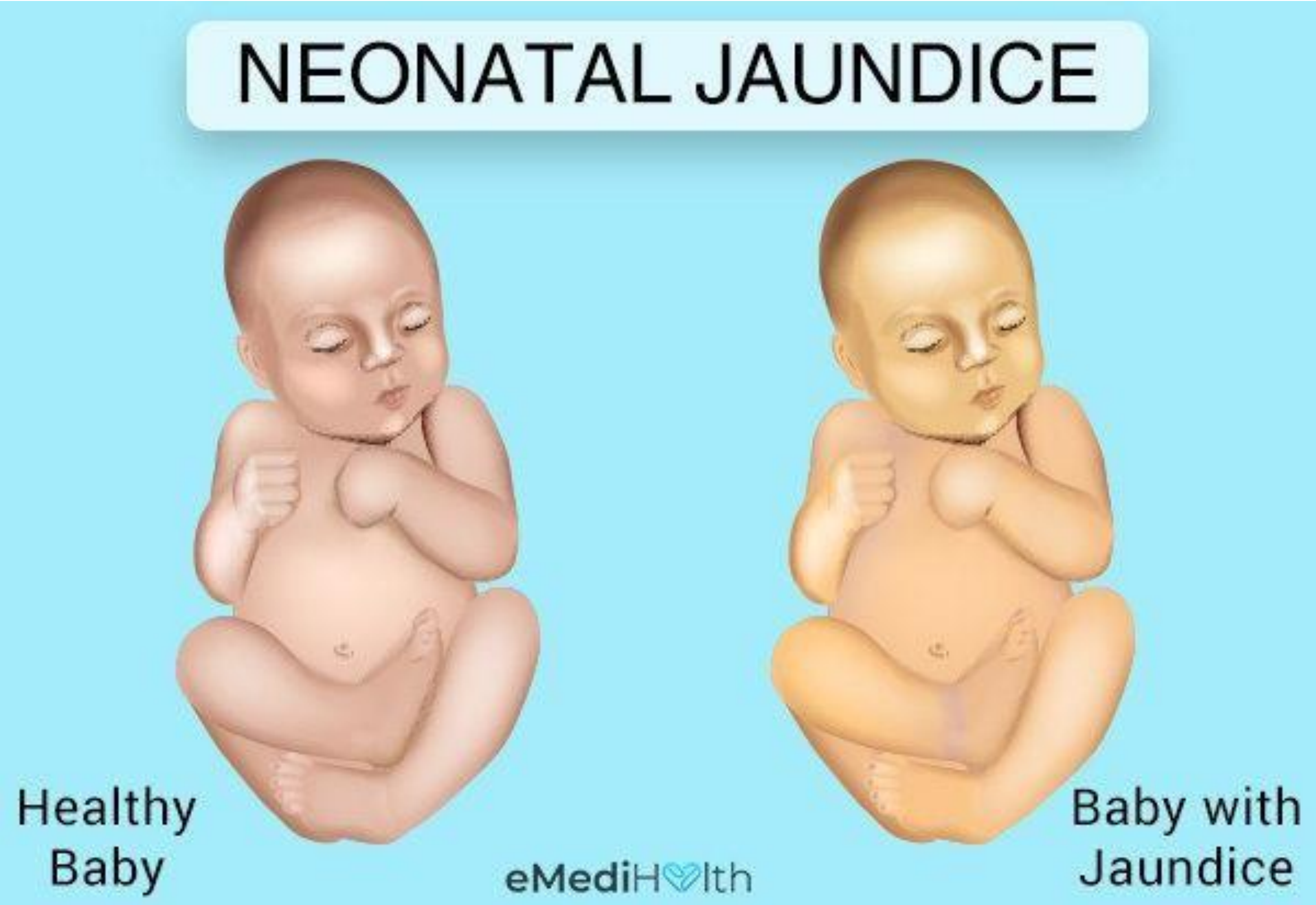


Figure 2:Healthy baby (left) and a jaundiced baby (right)

Bilirubin is a natural bi-product from the breakdown of red blood cells (Falciglia, 2016). The liver plays the role of converting unconjugated bilirubin to conjugated bilirubin. Unconjugated bilirubin is simply bilirubin that is 'free', it is not bonded or bound. It is very dangerous when in high amounts; it becomes very toxic. New-borns have immature livers that are unable to successfully convert unconjugated bilirubin to conjugated bilirubin. Due to this, there is a build-up of bilirubin in the blood which will result in jaundice. Severe hyperbilirubinemia can cause serious brain damage, deafness and death, a medical condition called kernicterus (Falciglia, 2016).

Another treatment modality used in the treatment of hyperbilirubinemia is phototherapy. It is usually the first treatment method applied, but when it proves ineffective, exchange transfusion is performed to save the life of the baby. *Figure 3* is a photograph showing phototherapy used in the treatment of hyperbilirubinemia.

Figure 3:Phototherapy used in treating hyperbilirubinemia

Haemolytic disease of the new-born (HDN) refers to the medical condition where there is the premature breakdown of the new-born's red blood cells because of incompatibility of the blood types of the mother and the baby. The mother's antibodies attack the baby's red blood cells and they are broken down and destroyed (a process called haemolysis). HDN causes enlargement of organs and hyperbilirubinemia. Hyperbilirubinemia occurs in this case too; because the breakdown of red blood cells produces bilirubin; hence, rapid breakdown will ultimately lead to hyperbilirubinemia. In this case, intravenous immunoglobin (IVIG) may be administered to reduce the breakdown of red blood cells (LK, 1947). However, an exchange transfusion will be needed to save the life of the baby if IVIG proves ineffective (Nader, 2019).

CHOICE OF ROUTES AND TECHNIQUES FOR THE EXCHANGE

Blood exchange transfusion (BET) may be performed with

(a) a single catheter in the umbilical vein

(b) a catheter each in umbilical vein and umbilical artery.

SINGLE-CATHETER PULL PUSH TECHNIQUE

This method is usually performed by one medical personnel. It is a traditional method, where only one single umbilical venous catheter is used (Giancarlo Liumbruno, 2009). The umbilical vein catheter is connected to two three-way taps or stopcocks that are connected in series and the end of this series connection is connected to a syringe. Blood is drawn in aliquots of 5-20mL depending on baby's size and replaced with exact volume of blood. This is because per the weight of the baby, if greater than a certain amount of blood is withdrawn, the baby may pass out, or develop some organ failures. Blood is also withdrawn at a rate of 1.5-2mL/kg/min. During the procedure a nurse records the blood volume drawn and administered for each cycle. Vital signals of the baby must also be recorded. *Figure 4* is an example of how one catheter 'push-pull' technique works.

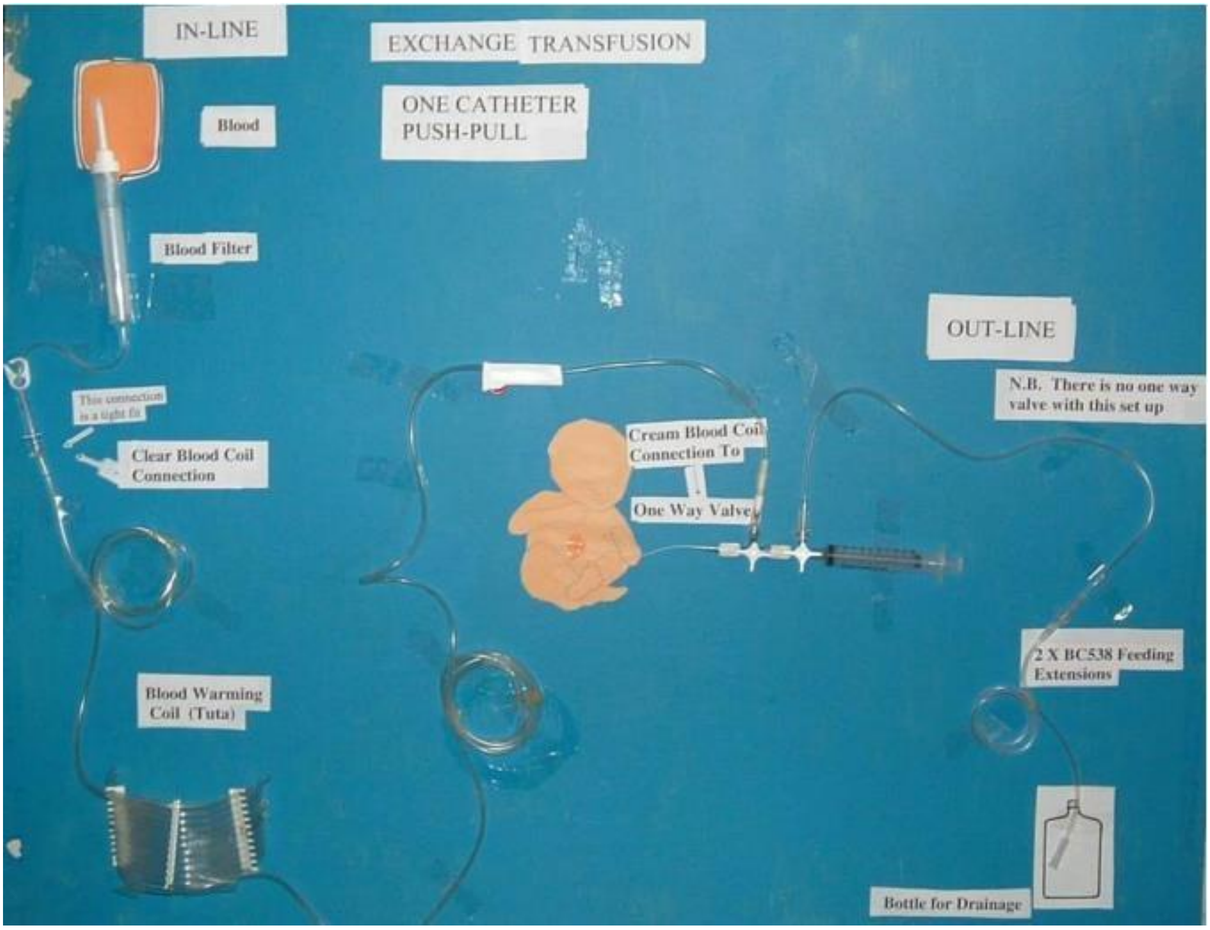


Figure 4: One catheter push-pull technique

A cycle in a single catheter push-pull method refers to a complete sequence of actions that is used to exchange a single calculated aliquot of blood.

A complete cycle comprises of:

- Slow and steady withdrawal of calculated quantity of blood from the baby.
- Discard blood withdrawn from the baby (fast)
- Draw equal volume of blood (taken from the baby) from the donor blood bag.
- Slow and steadily administer donor's blood into the baby.

There are standard aliquots of blood tolerated for exchange transfusion. These include the following in Table 1:

Table 1: Standard aliquots of blood for exchange transfusion

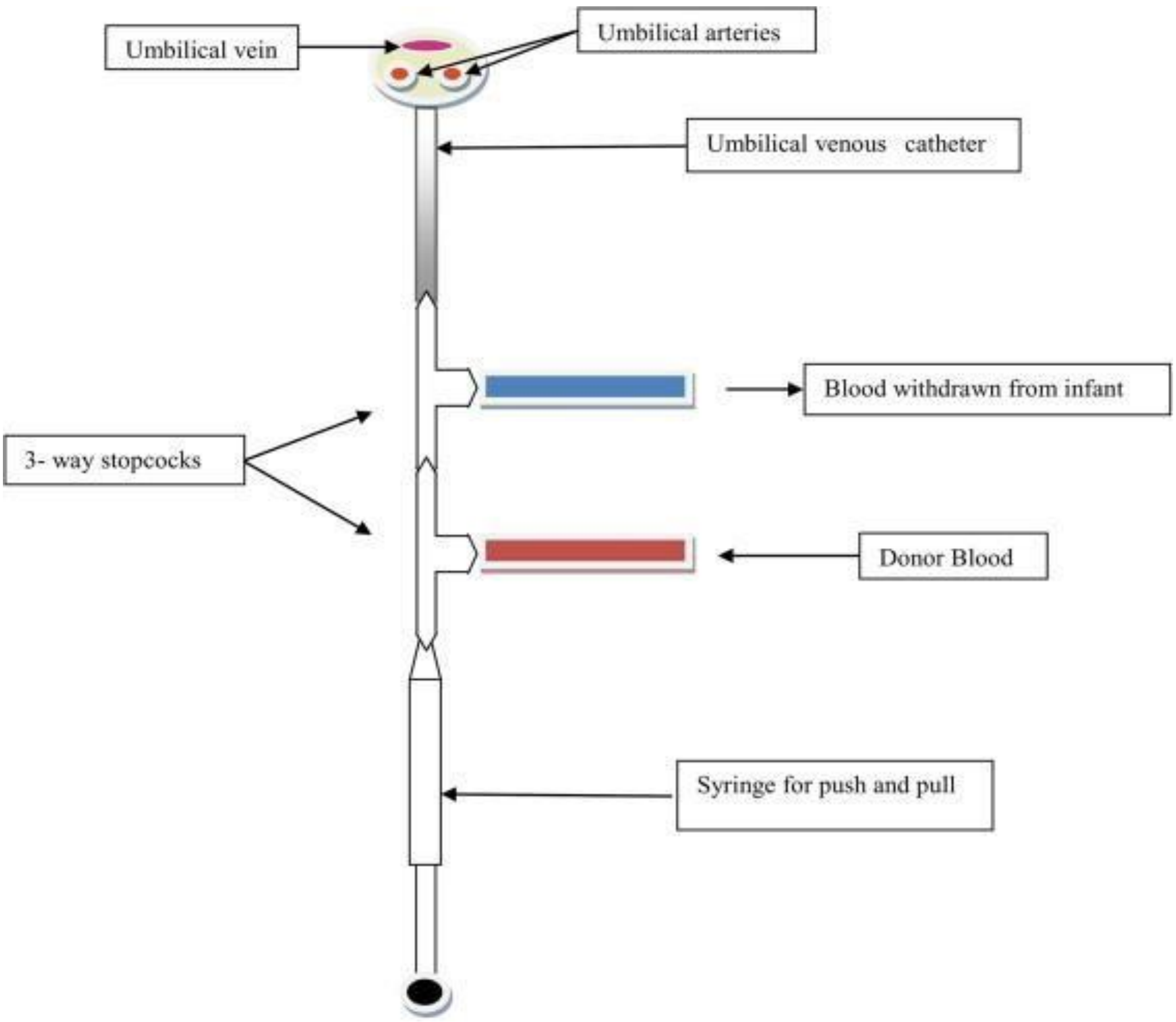


Figure 5: Diagrammatic representation of connections in one catheter push-pull technique

Mass of the baby	Volume for transfusion
< 1500g	5ml

1500g-2500g	10ml
2500g-3500g	15ml
>3500g	20ml

Figure 5 above is also showing a diagrammatic representation of connections of a catheter push-pull technique.

THE BASIC STEPS FOR MAKING THE SET-UP IN FIGURE 5 ABOVE IS AS FOLLOWS:

1. Umbilical venous catheter is fixed into the baby's umbilical vein
2. Two three-way taps are connected in series as shown in figure 6 below
3. The two three-way taps are attached to the umbilical venous catheter as shown in figures 5 and 6
4. Blood discarding line is connected to the side port of the three-way tap closest to the baby (the collection bag should be below bed level as shown in figure 4)
5. A 10 or 20ml syringe is connected to the port of the three-way tap furthest from the baby as shown in figure 4.

6.Donor blood line is connected to the side port of the three-way tap furthest from the baby

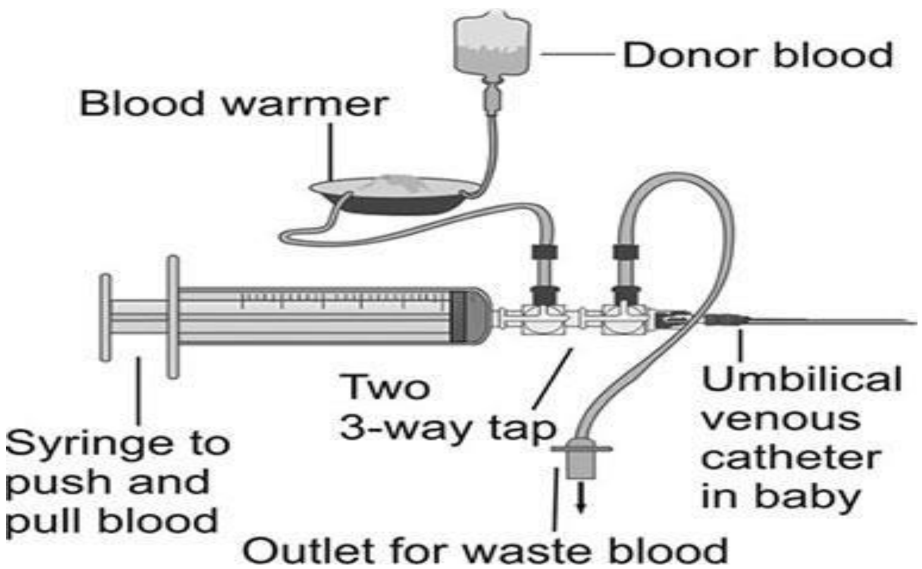


Figure 6: Two three-way valves connected in series

DOUBLE-CATHETER PULL PUSH TECHNIQUE

Another name for double-catheter pull-push technique is the Isovolumetric method. This process is performed by two people. Here equal aliquots of blood are withdrawn and administered simultaneously through different blood lines. The blood lines used are umbilical vein and umbilical artery. Blood is taken from the baby through the umbilical artery and equal volume of blood is simultaneously infused into the baby through the umbilical vein. Note that arterial lines are mostly for withdrawals and not for administering donor blood, this is because arterial blood pressure is much higher than venous blood pressure. The method we want to dwell on for our project is the single catheter push pull, this is because the double catheter push pull method is complex and needs to be performed by two people. The single catheter push pull method is simple and mostly used by doctors or medical personnel in our setting.

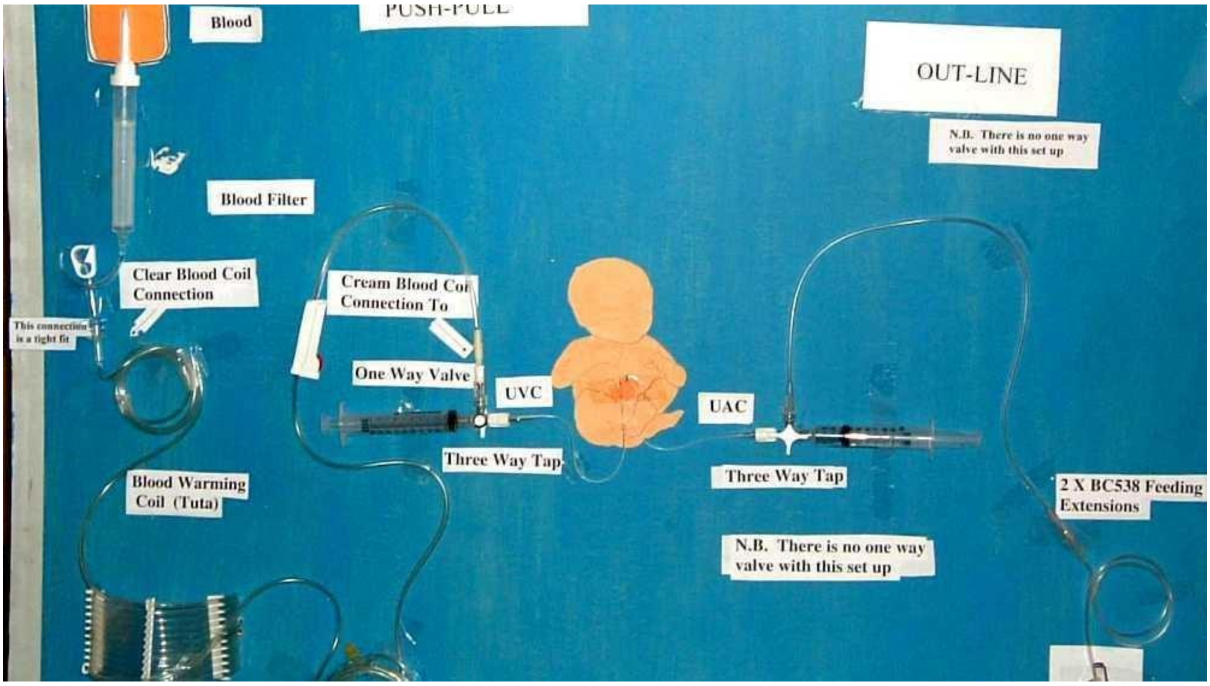


Figure 7: Double catheter push-pull exchange transfusion technique

Blood volumes drawn and injected into the baby are recorded after every cycle during the exchange procedure. Also, the baby is monitored closely and vital signs are recorded at an interval of 15-minutes (Nader, 2019). The vitals checked include:

- The heart rate
- The heart rhythm
- The blood pressure
- The temperature
- The respiratory rate
- The oxygen saturation
- The respiratory effort
- The infant's colour, tone, and behaviour. Etc.

BLOOD VOLUME CALCULATIONS

The problem with estimating the volume of exchange blood in babies has to do with the volume required to achieve a desired increase in the level of haemoglobin in the babies. A formula has however been established based on the assumption that the increase in haemoglobin concentration during the course of the procedure is a linear function of the volume exchange (Brian Berman, 1979). The increase in change in haemoglobin concentration during exchange transfusion is greatest during the early phase and progressively diminishes during the course of blood exchange.

The table below is used to explain the calculations involved in delivering aliquots of blood per cycle until the total volume is delivered.

The exchange volume parameter is given as:

$$\frac{Total\ volume\ to\ be\ delivered}{1.0\ kg}\ x\ weight\ of\ baby$$

Table 2: Total exchange volume parameter

	Weight	Exchange volume parameter	Total volume of blood
Preterm	$1kg$	$\frac{200ml}{1.0kg}\ x\ 1.0kg$	$200ml$
Term	$2kg$	$\frac{160ml}{1.0kg}\ x\ 2.0kg$	$320ml$

The *Table 2* above explains that a total of $200ml$ and $320ml$ will be administers to preterm and term babies respectively during the exchange procedure. This calculation is based on the exchange volume parameter. After the total exchange volume is determined, the number of cycles and duration of the exchange process is subsequently calculated using the formulae below

$$Number\ of\ cycles = \frac{\text{Total exchange volume}}{blood\ volume\ per\ cycle}$$

$$Duration = Number\ of\ cycles\ x\ time\ taken\ per\ cycle$$

Table 3: Exchange cycle and duration for both preterm and term babies

	Aliquots of blood per cycle	Number of cycles	Duration
Preterm	$5ml$ per cycle	$\frac{200ml}{5ml} = 40\ cycles$	$40\ cycles\ x\ 4\ mins = 160\ mins$
Term	$10ml$ per cycle	$\frac{320ml}{10ml} = 32\ cycles$	$32\ cycles\ x\ 4\ mins = 128\ mins$

Table 3 shows that at a rate of $5ml$ per cycle, it will take $40\ cycles$ to deliver a total of $200ml$ of blood during the exchange procedure. Subsequently, it will take a total of $2hour$ 40 $minutes$ for the whole procedure. Similarly, it will take $32\ cycles$ to deliver a total of $320ml$ of blood at a rate of $10ml$ per cycle to a term baby. This process will span $2hours$ $8minutes$ (Gluzman, 2019).

COMPLICATIONS OF EXCHANGE TRANSFUSION

Although exchange transfusion is considered a safe and effective procedure it is not without complications (refer to Table 4). Its mortality rates are between 0.5% and 3.3%, hence, current recommendations for carrying out an exchange transfusion are based on the balance between the risks of encephalopathy and the complications related to the procedure. However, most of these complications can be avoided if the procedure is carried out very slowly, with the right hygiene and care. Often the best management of these complications is to slow down or pause the exchange (Safer Care Victoria, 2018).

Table 4: Complications that could occur during exchange transfusion (left) vs those that can

Complications during the procedure	Complications after the procedure
Air Embolus	Infection
Volume Imbalance	Thrombocytopenia
Hyperkalaemia	Hypoglycaemia
Arrhythmias	Blood transmitted infections
Respiratory Distresses	Hypocalcaemia
Anaemia/ Polycythaemia	Anaemia/ Polycythaemia
Acidosis	Hypernatremia

CHAPTER THREE

METHODOLOGY

This chapter discusses the data acquisition and analysis process, tools and materials used to design the automated neonatal exchange transfusion device

3.1 DATA COLLECTION

The data collection process involved the use of both primary and secondary data. Phone call conversations were held as part of primary data. Secondary data on the other hand was obtained by reading articles, journals and watching YouTube videos on content related to the following topics:

I. Hyperbilirubinemia
II. Exchange transfusion.
III. Mechanics of motors, nuts and threaded rods
IV. Valving technology
V. Bubble trap technology among others

3.2 DATA ANALYSIS

Data was primarily analysed using qualitative approach. Brainstorming sessions, coupled with reading of previous literature guided the analysis approach.

3.3 DESIGN PROCESS

The goal of this project is to combine mechanics, electronics and programming to control the entire exchange transfusion set-up. To do this, each segment is carefully worked on while considering their overall performance in collaboration with the others. A bipolar stepper motor is required to provide high torque and precision for moving the plunger of the syringe. Two unipolar stepper motors were also used for providing moderate torque and good precision for turning the three-way valves. Bipolar stepper motors provide rotatory motion, however, for our device to function effectively, the plunger will have to be pulled or pushed in a linear motion. Therefore, a coupler and a guiding rod are used to translate the rotary motion of the motor in linear motion. To do this, a motor coupler is interfaced with the motor at one end and a threaded rod at another end. The guiding rod is then used to hold a pusher block in position such that only linear motion is permitted.

The main segments were the electronics, mechanics and programming segments. There were sub divisions for some segments.

3.4 Mechanics

Mechanics was sub-divided into push-pull and valve control segment.

3.4.1 Push and pull segment

In this segment, we focus on replacing the manual drawing and delivering of blood with a bipolar stepper motor, lead screw and guiding rod.

The stepper motor rotates clockwise. Its rotation causes the threaded rod to rotate in the same direction because of its attachment to the shaft of the motor. The pusher block which is attached to the rod propels forward in a linear motion. The linear motion is achieved with the help of the guiding rods as it restricts the pusher block from moving sideways. As the pusher block moves forward, it pushes the plunger of the syringe which causes a pumping action.

To cause a sucking action, the motor rotates counter clockwise which causes the threaded rod to rotate in that regard. Because the mini connector of the pusher block holds the syringe tightly, it pulls the plunger of the syringe along which causes a sucking action as it moves backward. The continuous rotations of the stepper motor and its direction causes us to suck and pump blood in and out of the baby.

Stepper motors

A stepper motor is a unique type of a DC motor that moves in distinct steps during its rotation. These distinct steps are characterized by a step angle. This characteristic of a stepper motor makes it suitable to incorporate it with digital systems such as a microcontroller. The following were some of the considerations made before settling on a bipolar stepper motor

- Stepper motors are relatively inexpensive as compared to other motor types. The cost depends on the choice of motor, the resolution required and the application at hand.

- Stepper motors can actually be used without any sensing device. The motor moves in distinct steps as defined by step angle. The only thing needed is to count the number of steps to position the motor accordingly. This kind of system is called the open loop system.
- The unique torque characteristic of the stepper motor makes it ideal for position applications. Stepper motors hold their position firmly at a given step providing a relatively high holding torque. There is no mechanical braking due to higher torque at low revolutions per minute.

1. Bipolar stepper motors

Bipolar stepper motors have two set of coils, two wires per coil giving it four connections. It has a very high torque at a slower maximum speed. It needs a very advanced controller circuit for controlling its movement.

PARAMETERS CONSIDERED FOR CHOOSING STEPPER MOTORS

Step angle: This is how much the shaft of the stepper motor advances per step current required.

Current: Parameter required to drive motor

Resistance: Relates to current and voltage specification.

Inductance: It is related to maximum speed

Holding torque: Amount of torque the motor can exert when it is energized.

Detent torque: It is the amount of torque the motor has when it is not energized.

Figure 8 below is an example of a stepper motor

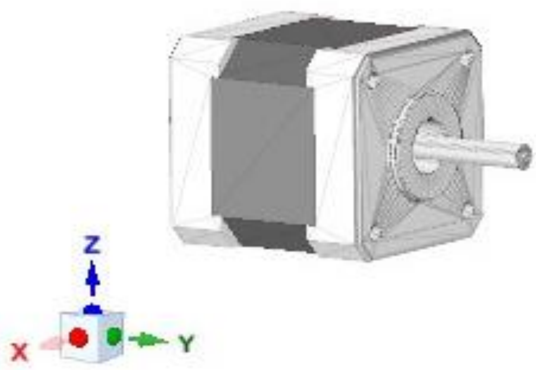


Figure 8: Stepper motor modelled in Ansys

2. Lead screw and guiding rod

A single-start thread has one continuous thread running along the body of the screw (Figure 9). They are usually cheap and commonly used. A multi-start thread consists of two or more intertwined threads running parallel to one another. The lead distance of a double start thread is twice that of the single start thread and a triple start thread has twice that of the single start thread.

Figure 9: Lead screw and guiding rod

<u>Stepper motor movement</u>

To determine the distance that the nut moves along the lead screw, a number of factors are put into considerations. They include:

- Step angle: Determines how many pulses are needed for one revolution
- Pulses: Together with the step angle, dictate the number of revolutions the stepper motor will engage
- Pitch: The distance between two crests on a lead screw

The details of the calculations are explained in the programming segment

3. Syringe

A 20ml syringe is used in this case because that is the maximum volume of blood that will be drawn or delivered, depending on the weight of the body as explained earlier.

3.4.2 VALVE CONTROL SEGMENT

The components involved in the valve control segment include two unipolar stepper motors, two three-way valves in series and catheter ends connecting to the donor blood bag, waste container and to the baby. The approach is to replace the manual switching (Figure 10) of valves by the physician with motors. The motors will be controlled by a program written to a microcontroller. The program synchronizes the push-pull segment and the valve control segment to successfully complete the processes of drawing out blood and replacing it with fresh donor blood.

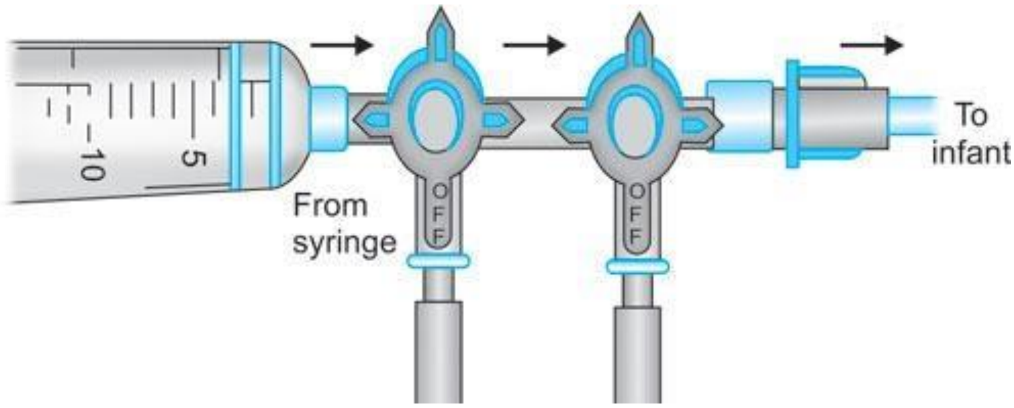


Figure 10: Manual representation of the valving system in exchange transfusion

4. Valves

The valves are used in controlling the direction of blood flow. They allow the movement or flow of blood in three different directions. Most of the automation focuses here since every cycle will require about 4 manual turns of the valves to achieve the required flow.

5. Unipolar motors

Unipolar stepper motors also have two set of coils with each having a centre tap. This motor can have six connections but they often have five as the two centre taps are tied together. It needs no voltage reversal for reversing direction. Unipolar stepper motors constantly apply voltage to the centre tap, wherever you attach the negative voltage will determine the direction the motor is going to run. It has a very low torque at higher maximum speed and it needs a very simple controller circuit.

3.5 ELECTRONICS

6. Power source

Currently, we use two power sources - 5v dc to power the Arduino microcontroller and a separate 6v dc to power the other components. Powering three stepper motors will require high voltage input but our design takes care of that by introducing step-up transformers (CN6009) where necessary. The input 6v is stepped up to 7v to power the two ULN2003 motor drivers interfacing the unipolar stepper motors. The output terminal of the CN6009 voltage booster is tapped and connected to the input of another CN6009 voltage booster where the voltage is stepped up to 12 volts to be used as an external power supply to the Arduino motor shield purposely for the bipolar stepper motor.

The transformers were incorporated into our design mainly to keep the input power requirement low. Also, the device can function on batteries when it is used in remote areas or in places where there is no electricity.

7. Motor drivers

The L293D is a 16-pin motor driver IC that is widely used. It is mostly used to drive motors, as the name implies. A single L293D IC can drive two DC motors at the same time, and the two motors' directions can be regulated individually. This IC is ideal for motors with an operating voltage of less than 36V and a current of less than 600mA that are to be controlled by digital circuits such as Op-Amps, 555 timers, digital gates, or even Micron rollers such as Arduino, PIC and ARM.

Stepper motors require pulse to move at specified number of steps, speed, direction and rate. A motor driver is responsible for sending these pulses. Each driver is selected specifically for the two types of motors used. To control the bipolar stepper motor, an Arduino motor shield was interfaced with the Arduino mega microcontroller. The movement of the two unipolar stepper motors on the hand was regulated by ULN2003 motor drivers.

The ULN2003 stepper motor driver PCB connects Arduino mega microcontroller to a stepper motor through a direct drive interface. The PCB has four inputs for connecting to the microcontroller, a power supply for the stepper motor voltage, an ON/OFF jumper, a direct connect stepper motor header and four LEDs to indicate stepping state.

8. Liquid Crystal Display (LCD)

LCD is the abbreviation for Liquid Crystal Display. It is a type of electronic display module that is utilized in a wide range of circuits and devices such as mobile phones, calculators, computers, television sets, etc. Multi-segment light-emitting diodes and seven segments are the most common applications for these displays. The primary advantage of utilizing this module is its low cost, ease of programming, animations, and the fact that there are no restrictions on displaying unique characters, special and even animations. The following are the primary features of this LCD;

I. The LCD's working voltage ranges from 4.7 to 5.3 volts.
II. It has two rows each of which can produce 16 characters
III. With no illumination, the current use is 1mA
IV. A 5x8 pixel box can be used to create any character
V. Backlight options include Blue and Green

I2C LCD ADAPTER

An 8-Bit i/o Expander chip-PCF8574- is at the heart of the adaptor. This chip transforms I2C data from an Arduino to the parallel data that the LCD display requires. The board also comes with a small trimpot to make fine adjustments to the contrast of the display. In addition, there is a jumper on the board that supplies power to the backlight, you can remove the jumper and apply an external voltage to the reader pin that is marked as 'LED'.

I2C module with LCD was chosen for our device because it uses less pins to connect to Arduino, hence leaving more pinholes for connection of other peripherals.

An integral part of our design is communicating with the user. To do this, we interfaced liquid crystal display (LCD) to prompt for inputs and inform the user about proceedings throughout the exchange transfusion procedure.

9. Keypad

Matrix keypads are used for inputting data. The 4x4 keypad has 8 pins of which 4 are row-pins and the other 4 are column pins. The Keypad was used in our design to enable us send inputs (usually decimal numbers representing the weight of the baby) to the microcontroller.

10. Switch

In order not for the device to always stay on (for energy conservation purposes) a switched was added to the design.

11. Arduino Mega

The Arduino Mega 2560 is an ATmega2560-based microcontroller board. It contains 54 digital input/output pins (with 15 of them capable of being used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It comes with everything you'll need to get started with the microcontroller; simply plug it into a computer with a USB cable or power it with an AC-to-DC adapter or battery. Most shields built for the Uno are compatible with the Mega 2560 board.

To interface and control all the other components, we needed microcontroller that had enough pins to accommodate each component and single-handedly control each of them. Also, we needed a controller with familiar programming interface and off-the-shelf packages. We resorted to the use of an Arduino Mega since it checks off all the requirements.

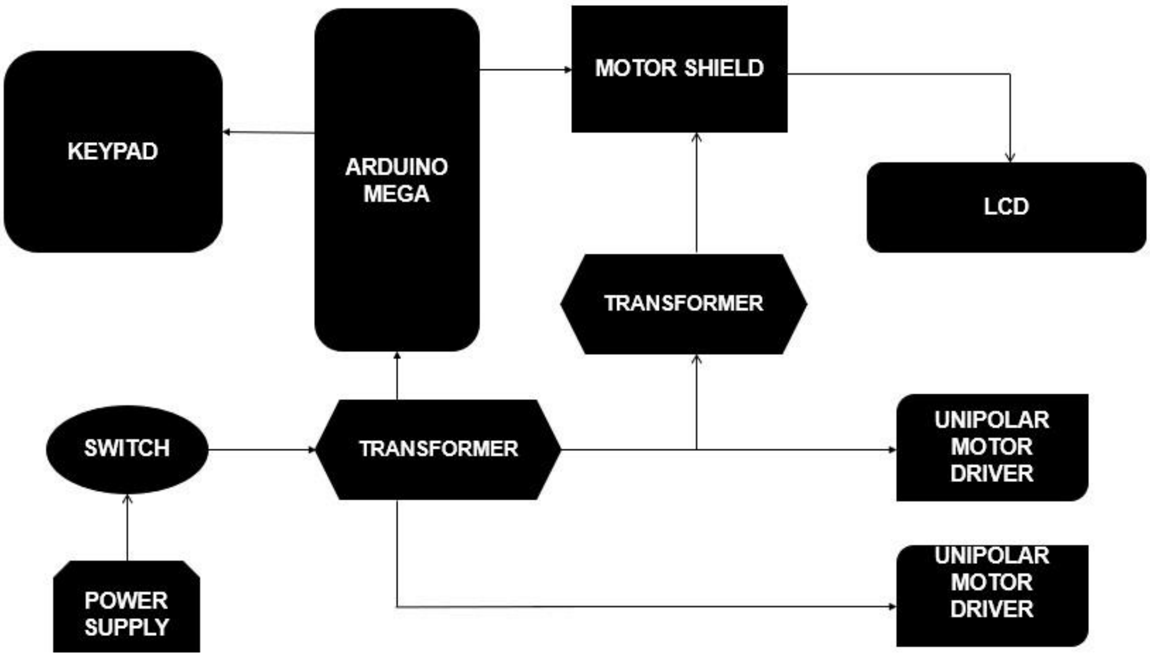


Figure 11: Block diagram showing connections between electronic components

Figure 11 shows the connections between the electronic components

3.6 DESIGN

The building of the physical model involved two main parts: electronics and mechanics.

Electronics

The following steps were followed in the combination of the electronics components to interface with the mechanic components.

1. Setting up main power supply.

A dc barrel plug connector is connected to the input pins of a transformer as shown in Figure 11. This serves as the main power supply for the entire device. Since we are concerned with designing a device that is energy efficient, we employ the use of transformers to step the small input voltage to a higher voltage to power the motors. The transformer is stepped up to 7V on the output terminal. This voltage is tapped to the input voltage terminals of the unipolar stepper motor drivers. Initially, dc batteries were used as the external power supply to the motor shield but this was replaced by connecting the output terminals of the first transformer to the second transformer and stepping it up to 12V for the bipolar stepper motor.

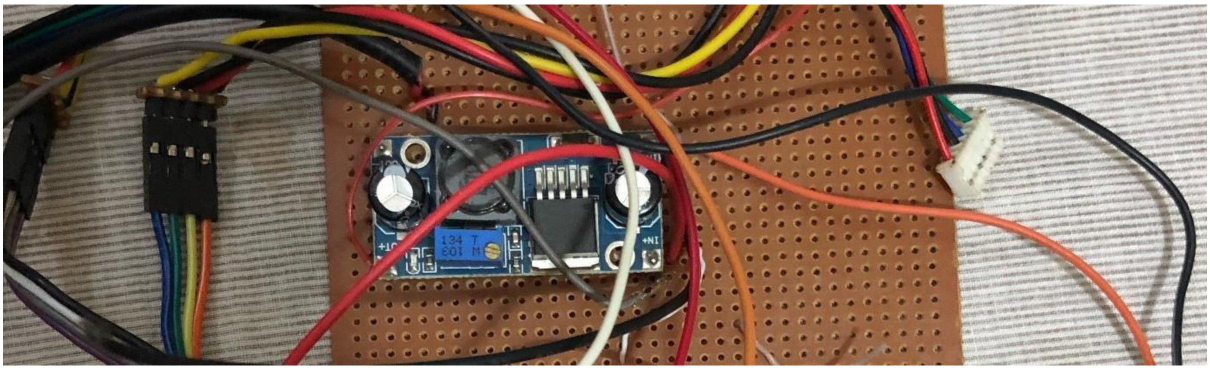


Figure 12: Setting up main power supply

Next, the Arduino mega was placed on the PCB board and the other components were connected to it as shown in *Figure 13* below.

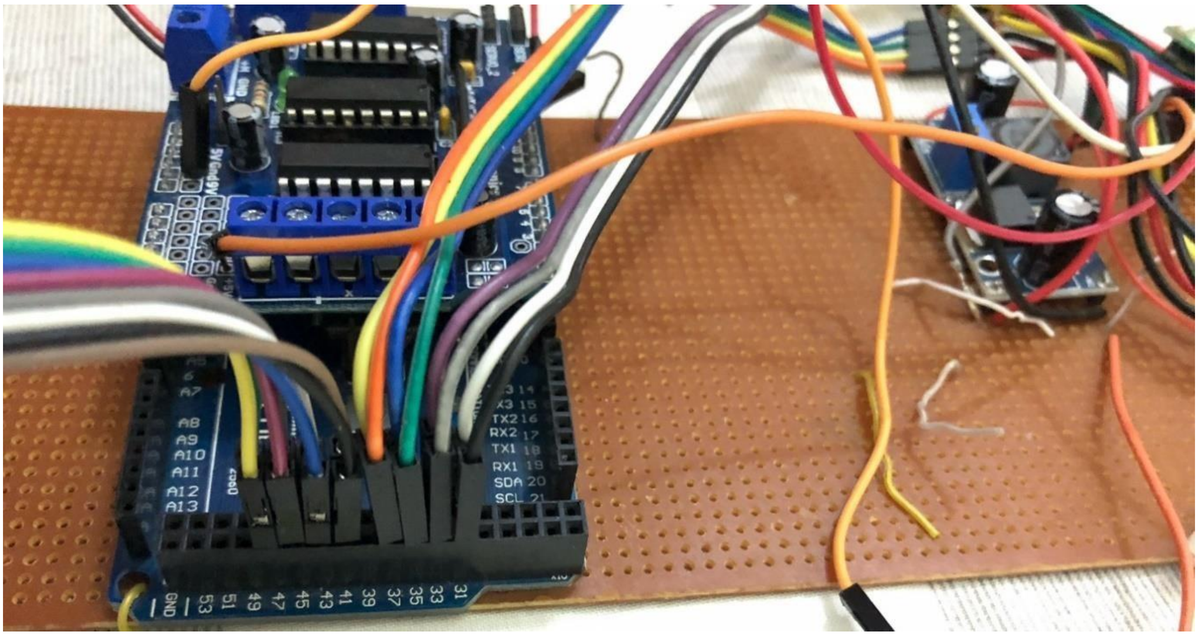


Figure 13: Connections on the Arduino Mega

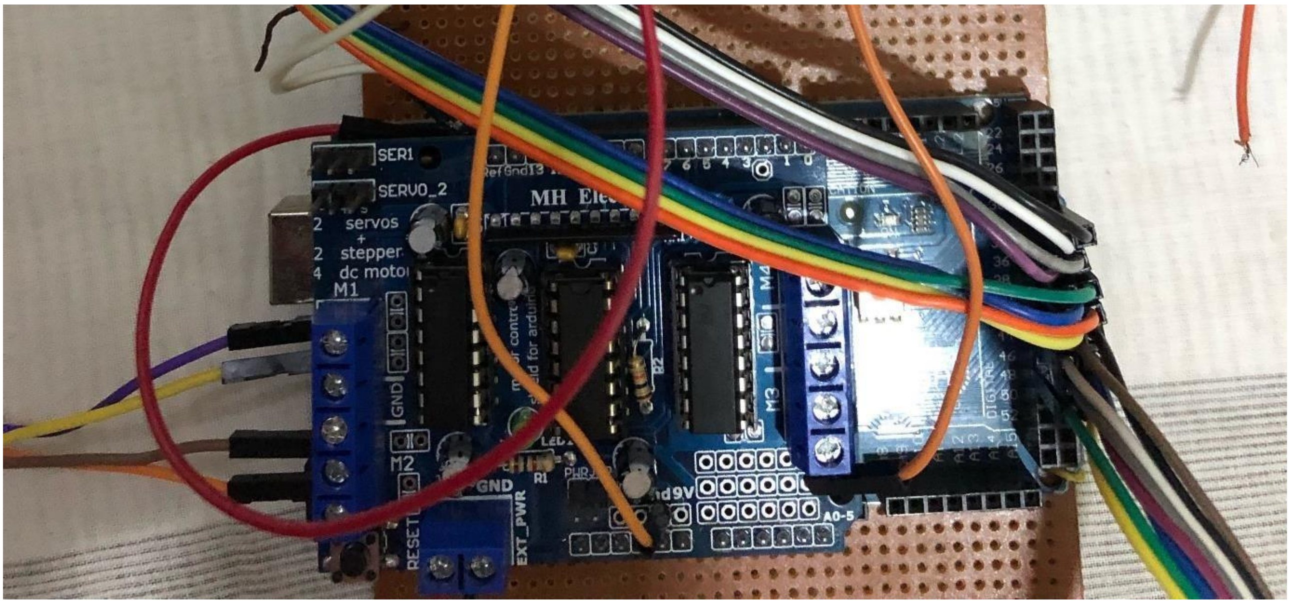


Figure 14: Interfacing motor shield with Arduino Mega

Figure 14 gives a clearer view of the Arduino Motor Shield sitting on the Arduino Mega. On the other hand, Figure 15 is a labelled picture of the electronic components connected for a test.

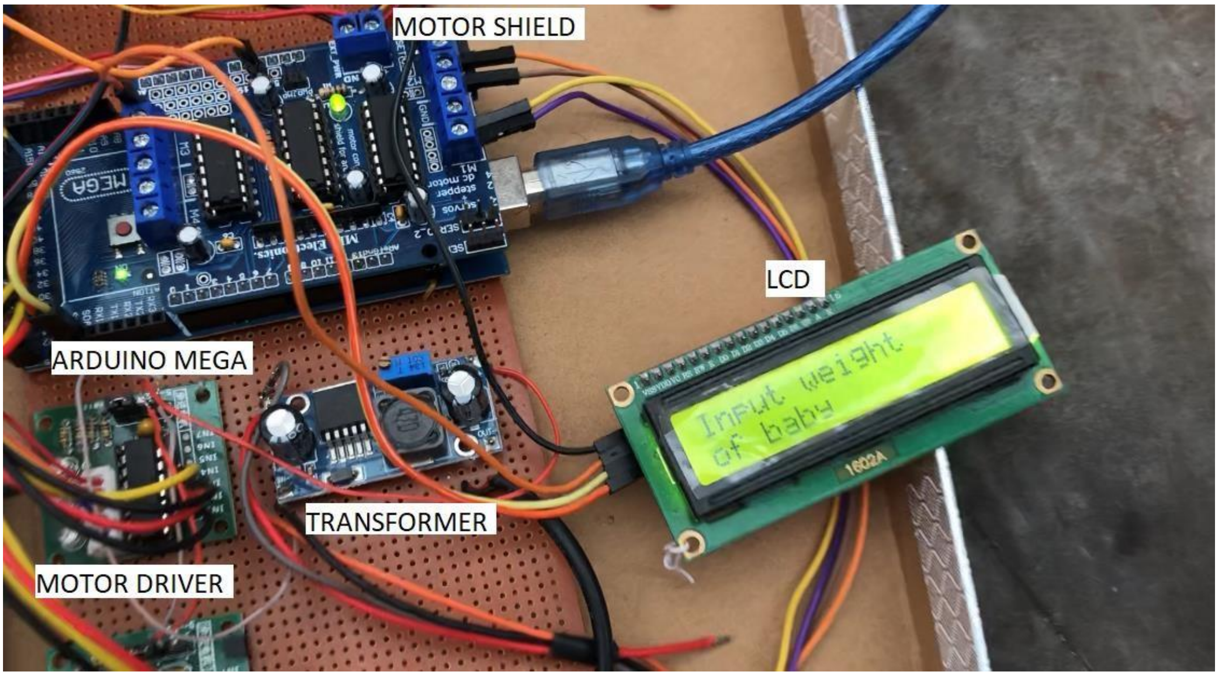


Figure 15: LCD displaying the code written to the Mega board

After testing the components to ensure they work, a component holder was designed to house the components. After considering the cost involved in 3D printing the component holder, we sought for a cheaper option- an off-the-shelf box.

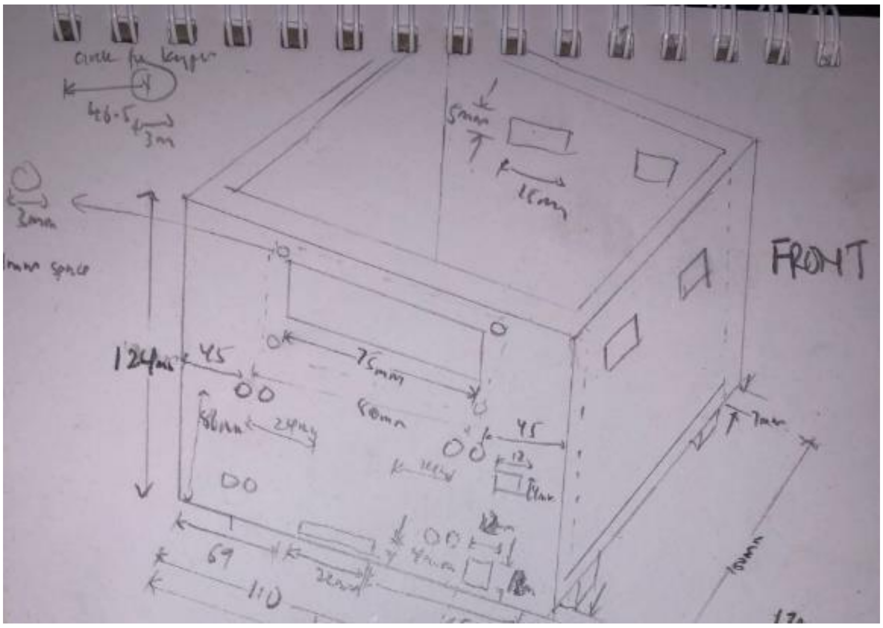


Figure 16: Specifications for component holder

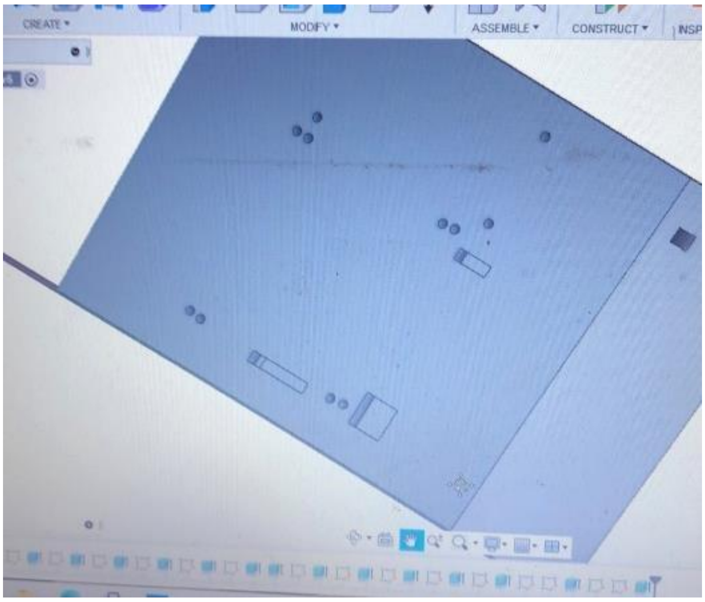


Figure 17: 3D model designed in Fusion 360

Figure 17 is a sketch of the container holder while Figure 30 is a 3D model designed in Fusion 360. Figure 18 below is the front view of the box with cut-outs to house the electronic components. The rear view of the box is shown in Figure 18 with the components labelled.

Figure 18: Off-the-shelf box used as component holder (front view)

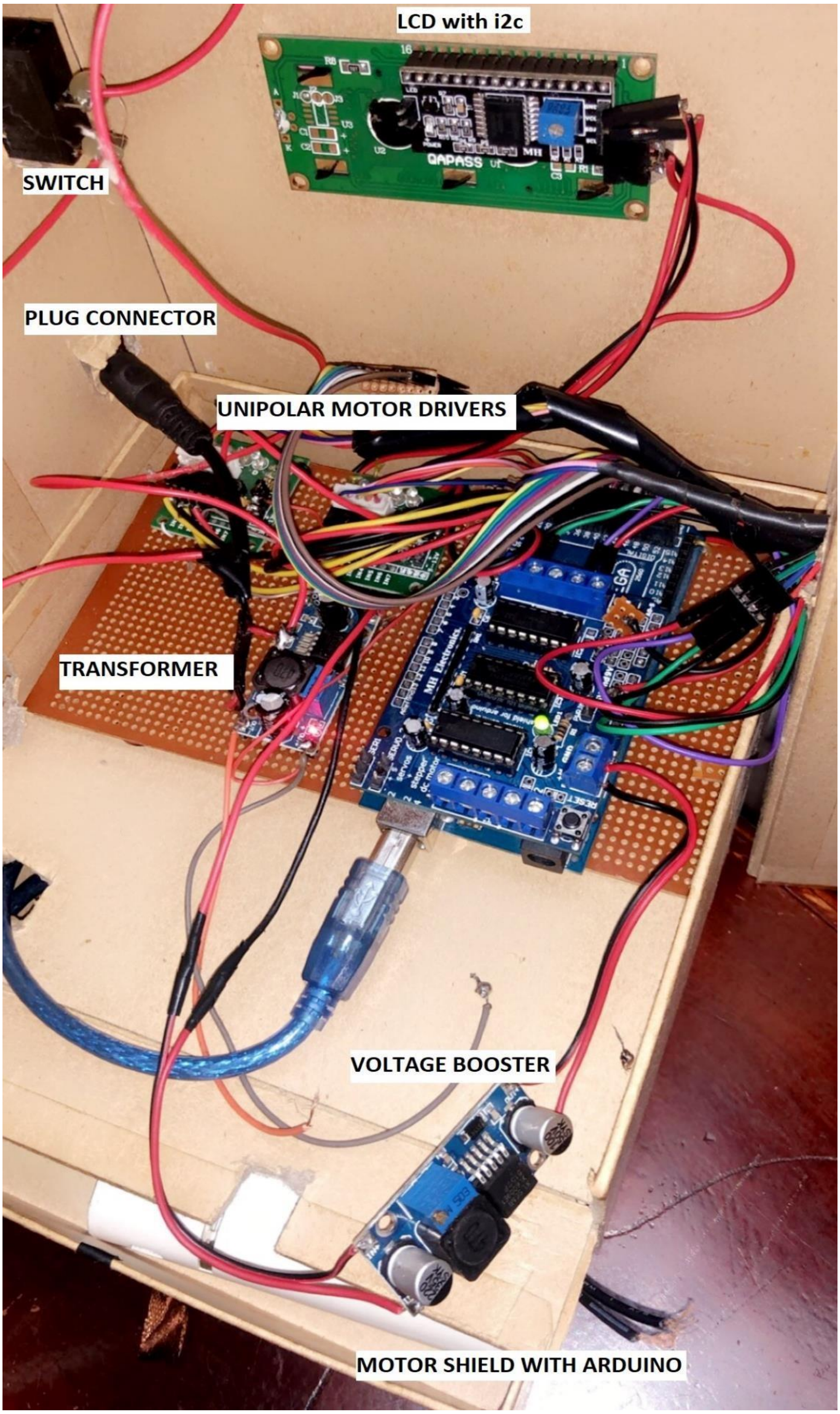


Figure 19: Off-the-shelf box used as component holder (rear view)

Mechanics

Before all the motors and rods are aligned on the base (as shown in Figure 19), the base model is carved out by a carpenter as shown in *Figure 20*.

Figure 20: Preparing base to accommodate mechanical components

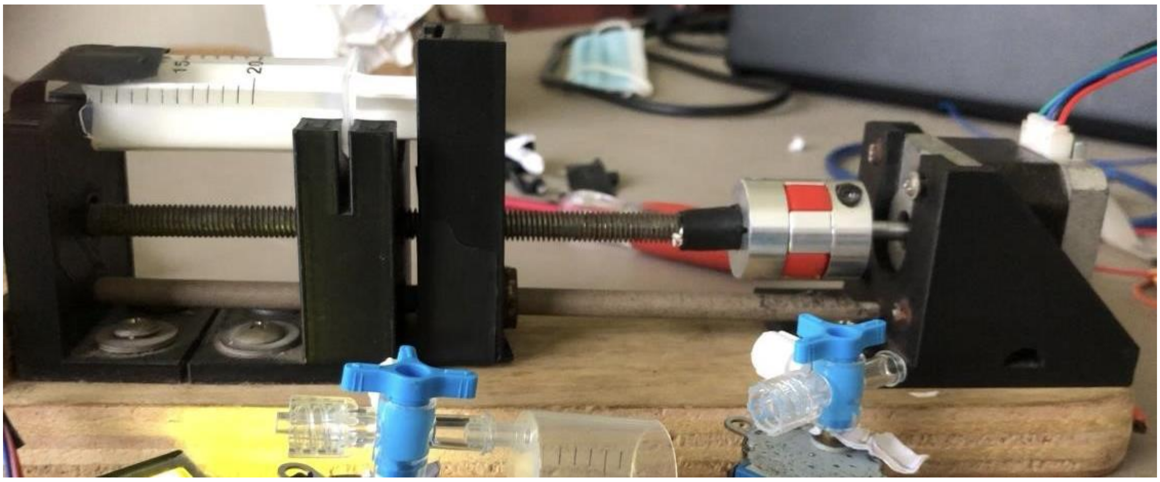


Figure 21: Aligning mechanical components on base

Figure 22 is a set-up for preliminary testing before components ae fitted into the main box. After the test, the components were arranged as shown in Figure 23. The cut-outs allowed for the LCD to be displayed, a switch to be fitted to the side and also for the wires connecting the mechanical components to be passed outside the box.

Figure 23: Components arranged in box and connected to the base model

Figure 22: Connecting mechanical components to the electronic components

3.6 Software implementation

The language choice for the programming was the Arduino-based C language. This was selected because it is one of the easiest programming languages used to write microcontroller programs.

The Arduino IDE was used to create the software for our device. The software features a back end that collects data such as the baby's weight. This is entered into the Arduino microcontroller using a keypad after which the various parameters required for exchange transfusion are calculated. These parameters include: total blood volume for the transfusion, time for drawing blood, time for delivering blood, total number of cycles for the transfusion, syringe volume, steps required to draw and deliver blood, speed of the motor for drawing and delivering blood. Variables were created to store values for the parameters listed above. Variables were named in such a way anyone can easily recognize the kind of data the variable will be storing. Variables for our software are shown in the snippet below:

```
int StepsRequired;
int count;
int data_count;
float babyWeight=0;
int syringeVolume = 15;
int totalBloodVolume;
int numberOfCycles;
int minutesPerCycle;
int number_of_steps;
int number_of_revs;
float rpmDraw;
float rpmDeliver;
char key;
char Data[3];
```

```
#include <AFMotor.h>
#include <Stepper.h> //Include the Arduino Stepper Library
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
```

Various imported libraries were needed for our Arduino to communicate with the other hardware components. The imported libraries are shown in the snippet below;

The <Stepper.h> library allows you to control both unipolar and bipolar stepper motors. The

<Wire.h> library allows you to control the I2C which is used in conjunction with LCD. The <LiquidCrystal_I2C.h> library passes information to the LCD. The <Keypad.h> library also helps to control the keypad.

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);


 lcd.init();                      // initialize the lcd
  lcd.init();


  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("ANET 2.0");
  lcd.setCursor(0,1);
  lcd.print("Albert & Kisser");
  delay(4000);

  lcd.clear();
  lcd.setCursor(5,0);
  lcd.print("WELCOME");

 delay(4000);
   lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Input weight of");
    lcd.setCursor(0, 1);
    lcd.print("baby in kg:");
```

Our program starts running from the void setup default function of the Arduino IDE. This inbuilt function runs the code once. To start with, the LCD is initialized using the 'lcd.init()' function which makes the LCD display. The first thing we display on the LCD is the name of the device and the version together with our names. The details are delayed for a period of four seconds and then the screen clears to print the 'WELCOME' message which also delays for four seconds. After the welcome message, the screen clears and displays 'Input Weight of the baby in kg'. The baby's weight prompt stays on the screen until the user of the device inputs the required data i.e., weight of baby. Figure 24 shows the prompt generated by the code snippet displayed above.

Figure 24: LCD of weight input prompt

```
void loop() {
  //Prompt user to input weight of baby in kg
 do{
    key = keypad.getKey();
    if(key){
    // Press key
    Data[data_count]='\0';
    Data[data_count] =  key;// store char into data array
   lcd.setCursor(12 + data_count, 1); // move cursor to show each new char
    lcd.print(Data[data_count]); // print char at said cursor
data_count++;
    }
   }while(data_count<3);

   lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Press # to ");
    lcd.setCursor(0, 1);
    lcd.print("continue");
 key = keypad.getKey();
    if(key='#'){

babyWeight=atof(Data);

getSyringeVolume();
ExchangeTransfusion();
    }
}
```

The code in the snippet above allows the user of the device to enter a key on the keypad. Here the do while loop is introduced because we want to make it possible for the user to be able to enter a decimal number. Each key pressed on the keypad is stored in an array. The size of the array is made to be equal to 3 in order so that we can save three characters that make up a decimal number. The do while loop starts with data_count=0, so as each key is pressed its current value is increased by 1 until its value is less than 3. Each key entered from the keypad is recognized as a string or character, therefore these characters need to be converted to floats in order for this value to be used in calculations of the various parameters. The *atof*, a special function in Arduino, is used for converting character value to float. Once the conversion is done, the value is stored in the *babyWeight* variable. There are two subfunctions in the *void_loop* function; these are the last two lines. These functions will be explained into detail in the next snippets.

```
//Function for getting Syringe Volume
int getSyringeVolume(){
  if(babyWeight < 1.5){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Use 5ml syringe");
    delay(5000);
    syringeVolume = 5;
    number_of_steps = 2941;
  }
```

```
else if(babyWeight>=1.5 & babyWeight <= 2.5){
  lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Use 10ml syringe");
    delay(5000);
  syringeVolume = 10;
  number_of_steps = 5883;
}
else if(babyWeight>2.5 & babyWeight <= 3.5){
   lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Use 15ml syringe");
    delay(5000);
```

First function we will talk about is the *getSyringeVolume()* function. This function helps to select a syringe volume (refer to Figure 23) and the number of steps it takes to drive the plunger in and out for a particular syringe volume. In the current hospital setting, 20 ml syringe is normally used for the exchange transfusion. We got one and we were able to determine the number of steps it will take to draw a 20ml blood volume. Based on the value we got for the number of steps for the 20ml syringe we used ratio and proportion to determine the number of steps it will take to draw blood using the 5ml, 10ml and 15ml syringes. The type of syringe to use is dependent heavily on the weight of the baby. After the baby's weight is entered, it enters straight into this function to actually check if the baby's weight is within a particular range and selects a syringe volume and the number of steps. A portion of the function definition is displayed in the snippet above. In the snippet, we can see there is an *if condition* that checks the range of the baby's weight. It then prompts the user to select a particular syringe volume.

This information is displayed on the LCD. If the baby's weight doesn't satisfy the first condition, it is passed on to the next condition till it satisfies a particular one. After the display on LCD, the variables *syringeVolume* and *numberOfSteps* are both initialized. These two variables are very key in the *ExchangeTransfusion*() function.

Figure 25: LCD showing user which syringe size to use

```
float ExchangeTransfusion(){

  //Pre-transfusion calculations
   float totalBloodVolume = babyWeight * 160;
   int numberOfCycles = ceil(totalBloodVolume/syringeVolume);
   float minutesPerCycle = 120/numberOfCycles;
   float minsForDrawing = ceil(0.4167*minutesPerCycle);
   //Minutes for drawing blood is 5/12 of the total minutes per cycle
   float minsForDelivering = ceil(0.5833*minutesPerCycle);
   //Minutes for drawing blood is 7/12 of the total minutes per cycle
   float number_of_revs = ceil(number_of_steps/200);
    int rpmDraw = number_of_revs / minsForDrawing;
    int rpmDeliver = number_of_revs / minsForDelivering;
```

The code snippet above shows the function definition of *ExchangeTransfusion()*, where all the necessary calculation needed for the exchange transfusion to take place is done. The first line of code in the function definition shows the calculation for the total blood volume needed for the transfusion. To arrive at this number, the baby's weight is multiplied by 160 because we are implementing the double blood volume procedure. To determine the number of cycles, the selected syringe volume divides the total blood volume for the exchange. It must be noted that all the functions are based on the baby's weight entered. The keyword *ceil* is approximate the value for the number of cycles calculated in case it is a decimal value. Next thing we calculated was the number of minutes each cycle in the transfusion process will take. A cycle in the transfusion process consists of successfully drawing of blood from the baby, discard it, drawing fresh donor blood and finally delivering the blood to the baby. For the minutes each cycle takes, we can see that 120 was divided by the *numberOfCycles;* the reason is because the process is supposed to be done within 2 hours and the number of minutes within 2 hours is 120. In the transfusion process, drawing of blood from the baby is a bit faster than delivering new blood to the baby. However, both processes are usually very slow. This has been explained in the earlier literature. A ratio was determined in order for *minutesPerCycle* to be shared between *minsForDrawing* and *minsForDelivering*. These ratios calculated were based on the observation of an actual exchange transfusion at HopeXchange hospital. For the observed transfusion, each cycle took a total of 6 mins. The time for drawing blood was 2 mins 30 seconds and time for delivering blood to the baby was 3 mins 30 seconds. On this basis, calculating for the ratios;

Ratio for drawing blood = $\frac{2.5}{6} = 0.41667$ and

Ratio for delivering blood = $\frac{3.5}{6} = 0.5833$

A stepper motor is responsible for moving the pusher block which also moves the plunger of the syringe in the drawing and delivering of blood. The stepper motor used for this project has a 1.8^{o} step angle. Some calculations were then made from this value.

Number of steps for one revolution is calculated as $\frac{360}{1.8} = 200$ steps

The speed at which the motor moves the pusher block is measured in revolutions per minute (rpm). Since the number of steps is specified for a particular syringe chosen, we need to calculate the number of revolutions that is within the number of steps specified. For example, if the number of steps specified is 2941 then;

Number of revolutions = $\frac{2941}{200} = 14.705\ revs$

Since the time taken to draw and deliver blood to the baby vary, the motor speed for each cycle phase will differ, resulting in different revolution per minute (rpm). Rpm's for both drawing and delivering need to be calculated by dividing the number of revolutions by *minsForDrawing* and *minsForDelivering* respectively. For instance, based on the number of revolutions we had in the previous calculation and the minutes per cycle for the live transfusion we witnessed;

Rpm for drawing blood = $\frac{14.705}{2.5} = 5.882\ rpm$

Rpm for delivering blood = $\frac{14.705}{3.5} = 4.20\ rpm$

```
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Total volume for");   //print to LCD
    lcd.setCursor(0,1);
    lcd.print("exchange:" );
    lcd.print(totalBloodVolume);
    lcd.print("ml");
    delay(5000);

lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Number of cycles:");
    lcd.setCursor(0,1);
    lcd.print(numberOfCycles);
    delay(5000);
```

The code above displays the total blood volume (an example is shown in Figure 24) and number of cycles after all calculations are done. One objective of this project is to make our model very communicative. This is important for the physician to manually calculate all the parameters and crosscheck that with our device. It also helps for the transfusion procedure to be stopped immediately the values calculated do not tally.

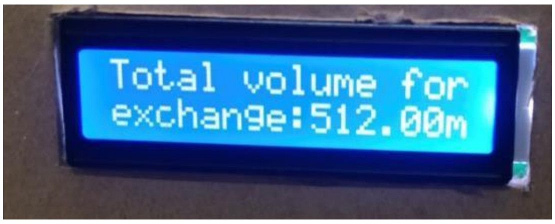


Figure 26: LCD showing total volume calculation

```
for(count = 1; count <= numberOfCycles; count++ ) {

   //STEP 1, DRAWING BLOOD FROM THE BABY
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Drawing blood");
lcd.setCursor(0,1);
lcd.print("from baby");
lcd.setCursor(10,1);
lcd.print("C" + String(count) + "/C" + String(numberOfCycles));
 motor1.setSpeed(rpmDraw);  // calculated rpm
 motor1.step(number_of_steps, BACKWARD, DOUBLE);
 //main motor moves backwards to draw blood from baby
```

Figure 27: LCD showing the procedure and cycle number

In this section of the code, we started a *for loop*, and this loop is responsible for performing the whole exchange transfusion process. The loop starts a *count* from 1 and performs the same set of instructions multiple times until the *count* value is greater than the number of cycles calculated in the previous section. The first section of the *for loop* displays ‘Drawing blood’ on the LCD (refer to Figure 25) and the motor direction is set to move backwards; causing the pusher block to draw blood into the syringe. At this point, the motor speed has been specified in the *rpmDraw* code which is already calculated for in the previous code. The *number_of_steps* value is also needed to give the distance the motor will move the pusher block to draw a specific volume of blood.

```
//STEP 2, DISCARDING BLOOD INTO WASTE BAG
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Discarding blood");
lcd.setCursor(9,1);
lcd.print("C" + String(count) + "/C" + String(numberOfCycles));

// Motor2 rotates in a clockwise direction to open valve to waste bag
StepsRequired  =  -STEPS_PER_OUT_REV / 3.5;
motor2.setSpeed(1000);
motor2.step(StepsRequired);


motor1.setSpeed(80);  // 10 rpm
motor1.step(number_of_steps, FORWARD, DOUBLE);  //main motor moves backwards to draw blood from baby
```

This code also displays 'Discarding blood' which is done after drawing blood from baby. This time, another motor comes into play (motor2), which is responsible for switching valve opening to waste bag and bad blood from the baby is discarded using motor1. This time the motor speed is set at 80 rpm, this is because discarding of blood is supposed to be done at a fast rate and 80 rpm was the highest speed our motor could move effectively without breaking. The motor direction is set to move forward to discard blood.

```
//STEP 3, DRAWING FRESH BLOOD FROM DONOR BAG
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Drawing fresh");
lcd.setCursor(0,1);
lcd.print("blood");
lcd.setCursor(8,1);
lcd.print("C" + String(count) + "/C" + String(numberOfCycles));

StepsRequired  =  STEPS_PER_OUT_REV / 3.5;
// Motor2 rotates in a counter clockwise direction to close valve to waste bag
motor2.setSpeed(1000);
motor2.step(StepsRequired);

StepsRequired  =  -STEPS_PER_OUT_REV / 3.5;
// // Motor3 rotates in a clockwise direction to open valve to waste bag
motor3.setSpeed(1000);
motor3.step(StepsRequired);
delay(200);

 motor1.setSpeed(80);  // 10 rpm
 motor1.step(number_of_steps, BACKWARD, DOUBLE);
 //main motor moves backwards to draw blood from baby
```

This section is responsible for drawing fresh blood from donor bag. Here *motor2* is moved counter-clockwise and closes the opening to the waste bag. Another motor (motor3) is moved clockwise to open way to the donor bag for fresh blood to be drawn. Drawing fresh blood from the donor bag is also drawn fast; therefore, the motor (motor1) speed is set at 80 rpm and is set to move backwards to draw blood from donor bag.

```
//STEP 4, DELIVERING FRESH BLOOD TO BABY
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Delivering blood");
lcd.setCursor(0,1);
lcd.print("to baby");
lcd.setCursor(10,1);
lcd.print("C" + String(count) + "/C" + String(numberOfCycles));

StepsRequired  =  STEPS_PER_OUT_REV / 3.5;
motor3.setSpeed(1000);
motor3.step(StepsRequired); // Motor3 rotates in a counter clockwise direction to close valve to donor bag

motor1.setSpeed(rpmDeliver);  // 10 rpm
motor1.step(number_of_steps, FORWARD, DOUBLE);  //main motor moves backwards to draw blood from baby
```

The section above is responsible for delivering fresh blood to the baby. *Motor3* is moved anticlockwise in order to close the opening to the donor bag. This time all openings point to the baby and fresh blood is delivered to the baby at a speed equal to the *rpmDeliver* and movement is also set to **'FORWARD'.**

```
//Prompt user
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Numb of complete");
  lcd.setCursor(0,1);
  lcd.print("cycles:");
  lcd.setCursor(8,1);
  lcd.print( String(count) + "/" + String(numberOfCycles));
    delay(5000);
```

This code here can prompt the user the number of cycles that has been completed in case the user comes back at any time during the process.

```
//Completion message
lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Transfusion ");
  lcd.setCursor(0,1);
  lcd.print("complete");
  lcd.setCursor(9,1);
  lcd.print(count);
    delay(5000);
```

The code above displays 'Transfusion complete' when *count* for the *for loop* becomes greater than the *numberOfCycles* value calculated.

CHAPTER FIVE
TESTING AND DISCUSSION OF RESULTS

5.1 TESTING PROTOTYPE

The device was set-up as shown below in Figure 26.

Figure 28: Final set-up for demonstration

Four separate transfusions were performed in accordance with the four weight divisions for exchange transfusion i.e., less than 1.5kg, 1.5kg to 2.5kg, 2.5kg to 3.5kg and finally weights more than 3.5kg. We manually calculated all the variables including the total blood for the transfusion, number of cycles and revolutions per minute (rpm) of the motor for each cycle among other parameters. Next, we entered the weights recorded on paper and cross-checked from our device to ensure that we were getting the same values. All the values were accurate and therefore we proceeded with the transfusion. A timer was set to confirm the time each cycle took. The bipolar stepper motor drew blood (in this case bisap) and discarded it. Bisap (sobolo) was used in place of water for all the demonstrations because it is pigmented like blood. Also, the liquid is mildly granulated and provides more friction compared to water.

The unipolar stepper motors coordinated the valve switching as programmed by the microcontroller. However, due to the strong torque needed to turn the valves, the switching was not as accurate as we expected it to be. A comparison between our model and the manual process was made.

The manual observation of the exchange transfusion process was observed at the HopeXchange Medical Centre. The baby's weight was 3.0kg and this was used to determine the total blood volume for the exchange, the number of cycles and minutes per cycle among other values.

From the observation, the following timestamps were recorded for five successive cycles. Clearly there is inconsistency in the time used for each cycle. Per the calculations, drawing blood is to take 2 minutes while delivering of blood is to take 3 minutes. On the other hand, ANET 2.0 recorded a total time of two minutes (+/- 5seconds) for each drawing session and a total time of three minutes (+/- 5seconds) for delivering blood. Table 5 shows the timestamps recorded during the observation of the manual exchange transfusion process.

Table 5: Timestamp for manual exchange transfusion

Cycle	1	2	3	4	5
Drawing Blood	2mins 36secs	2mins 26secs	2mins 38secs	2mins 17secs	2mins 22secs
Discarding Blood	6secs	6secs	11secs	8secs	7secs
Fresh Donor Blood	8secs	10secs	9secs	10secs	8secs
Delivering blood to baby	2mins 52secs	3mins 51secs	4mins 30secs	3mins 59secs	3mins 30 secs

Table 6: Comparing the manual process to the Automated process

Parameters	Manual Exchange Transfusion	Automated Exchange Transfusion
Pre-transfusion calculation time	Approximately 5mins	A few seconds
Tracking procedure	Manual documentation in log sheet	Automated and updated throughout the procedure
Precision	Less accurate	More accurate
Labour requirement for procedure	Two physicians	One physician
Attention requirement	Constant attention throughout the procedure	Flexibility of not observing the entire procedure
Time taken for discarding blood and drawing fresh donor blood	Relatively faster	Relatively slower

Table 6 shows comparison criteria used for checking both manual and automated exchange transfusion systems. The first is the pre-transfusion calculation time. In the hospital, the physician takes time to calculate parameters like the total blood volume required, the number of cycles, the minutes each cycle will take and the time each step within a cycle will last. These values need to be accurate as they may affect the baby negatively, should the computations be wrong. Our device is designed to perform all these computations in a matter of seconds. Also, there is no worry if the computations are wrong or not since accuracy is more guaranteed. Next, there is the need for the physician to check each step of the procedure in a log book in order not to get confused or miss a step. This is avoided with our model. The LCD displays each step and informs the physician how many cycles have been executed out of the total number. A major comparison factor was the precision in repetition. While there were significant inconsistencies recorded in the manual process, our device had no inconsistencies throughout an entire process. One aspect where the manual process beat our device was the time taken to discard was blood and draw fresh donor blood. This was because we wanted to maintain accuracy and there did not increase speed past 80rmp. As such, our device lags behind in terms of extra time used for the aforementioned. This over time lag contributes to a relatively longer procedure.

5.2 RESULTS

Before building a physical model, simulations were made to test the feasibility of the design. Ansys Mechanical Model was used to perform stress analysis on the base model and the threaded rod. After building the actual model, series of tests were run to analyse the efficiency of the device. These are explained in details in this chapter.

5.3 ANALYSIS OF BASE MODEL

All other components of the device rest on the base. It also bears the stress from screws used to fasten the motors and 3D printed materials. The material specification for the base is

Engineering Data: Material View

Plywood

Density	748 kg/m³
Structural	
Isotropic Elasticity	
Derive from	Young's Modulus and Poisson's Ratio
Young's Modulus	6.32e+09 Pa
Poisson's Ratio	0.245
Bulk Modulus	4.1307e+09 Pa
Shear Modulus	2.5382e+09 Pa
Isotropic Secant Coefficient of Thermal Expansion	6.93e-06 1/°C
Tensile Ultimate Strength	5.61e+07 Pa
Tensile Yield Strength	3.81e+07 Pa
Thermal	
Isotropic Thermal Conductivity	0.324 W/m·°C
Specific Heat Constant Pressure	1680 J/kg·°C
Electric	
Isotropic Resistivity	1.1e+06 ohm·m

Geometry | Engineering Data: Material View

Figure 29: Material specifications for base model

shown in Figure 29.

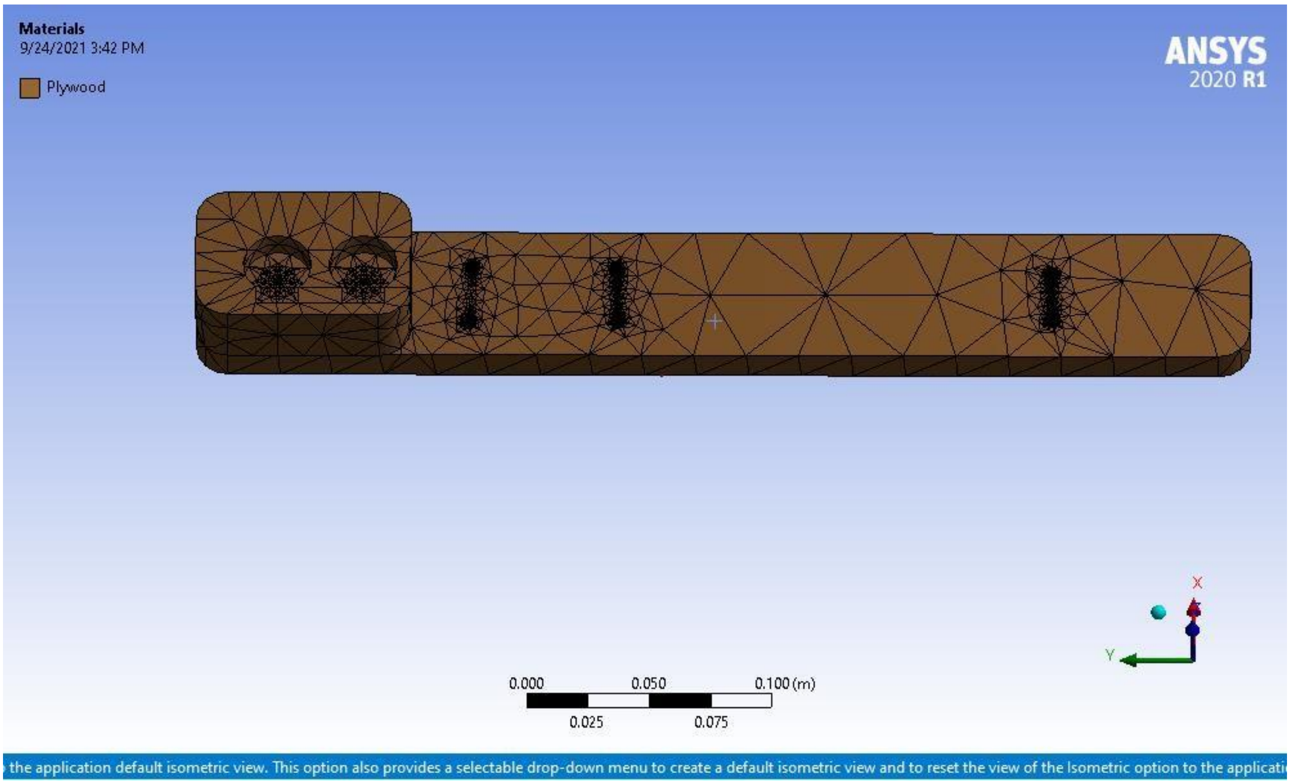


Figure 30: Plywood representation for analysis

Figure 30 shows the plywood material on the 3D model before analysis while figure 31 shows the rigid supports used on the device before the application of forces.

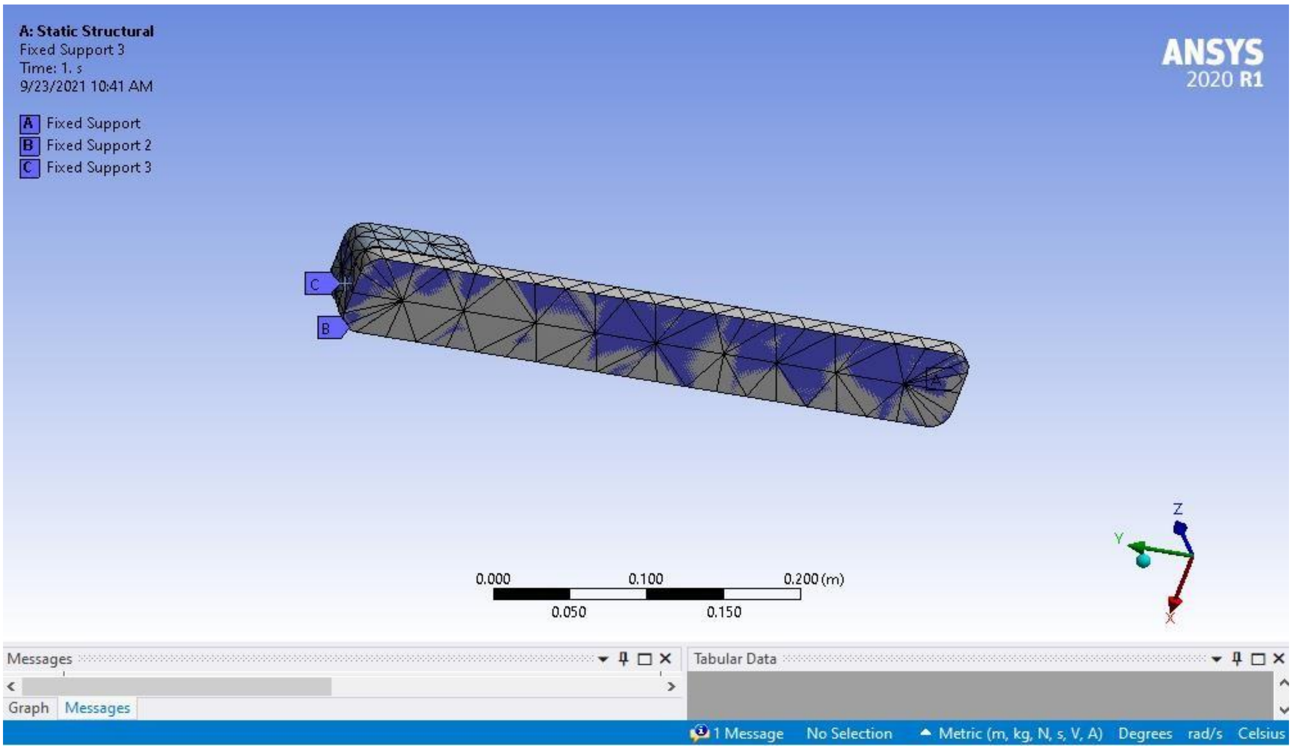


Figure 31: Fixed supports on base model

In Ansys, before forces are applied, fixed supports are used to restrict motion in certain directions for the analysis. After fixed supports are put in place, forces are applied to the surface of the board (refer to Figure 32). There are three forces applied. They are: a 3.43N force to the main board, 5.88N force applied in the unipolar stepper motor holder and a force of 1.663kN applied at screw holes. The results of these forces are analysed in the subsequent pages.

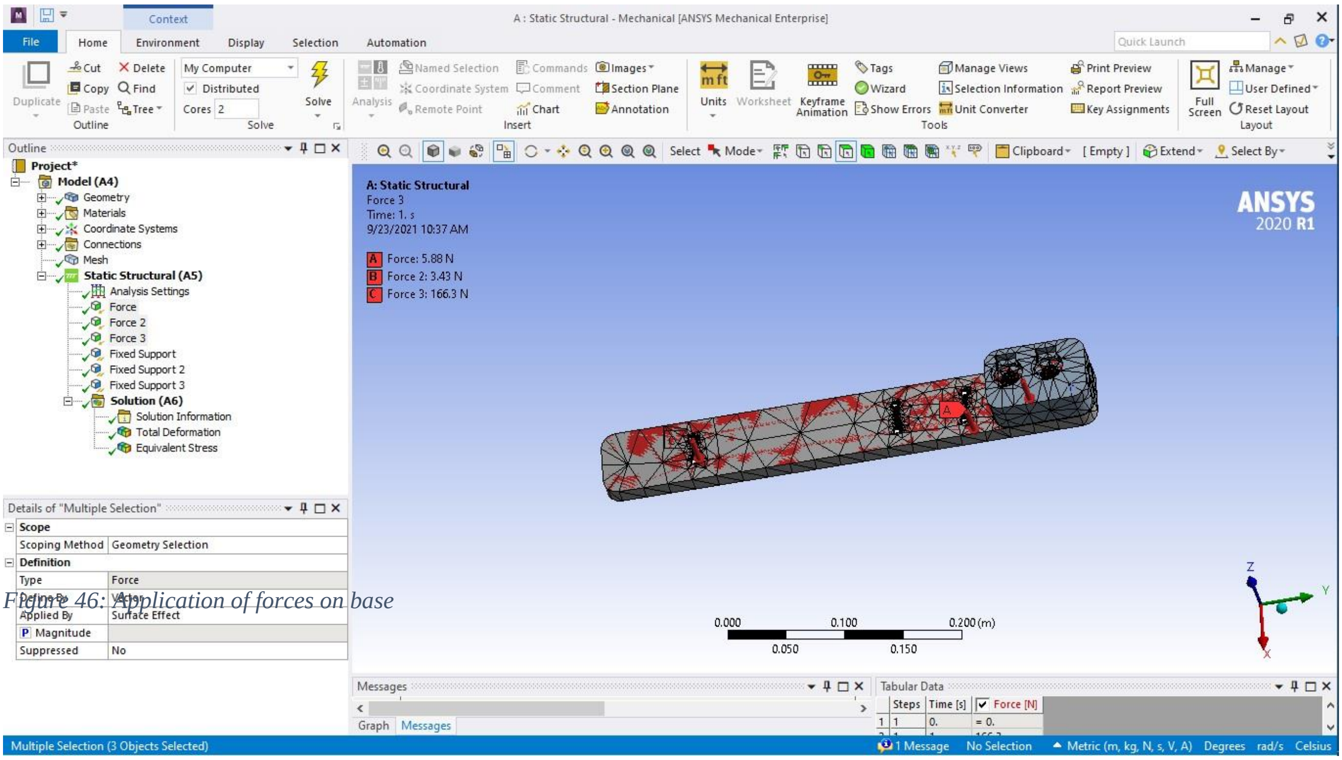


Figure 32: Force application on base model

For each application of force, total deformation of the model and the equivalent Von mises stress were accounted for (refer to Figure 33 below).

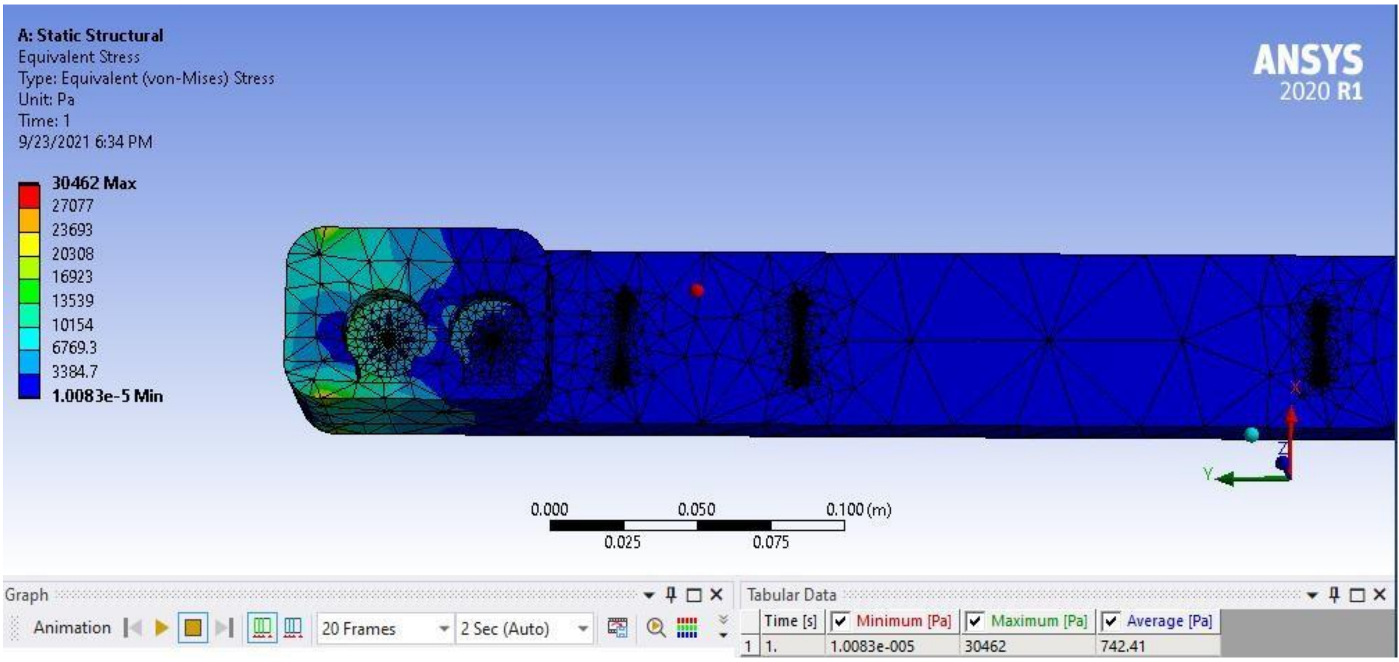


Figure 33: Equivalent (Von-Mises) stress of motor holder compartment v

From the simulation, the maximum stress is 30462Pa and it occurs at both the edges of the holes and the top base model as shown in figure 34)

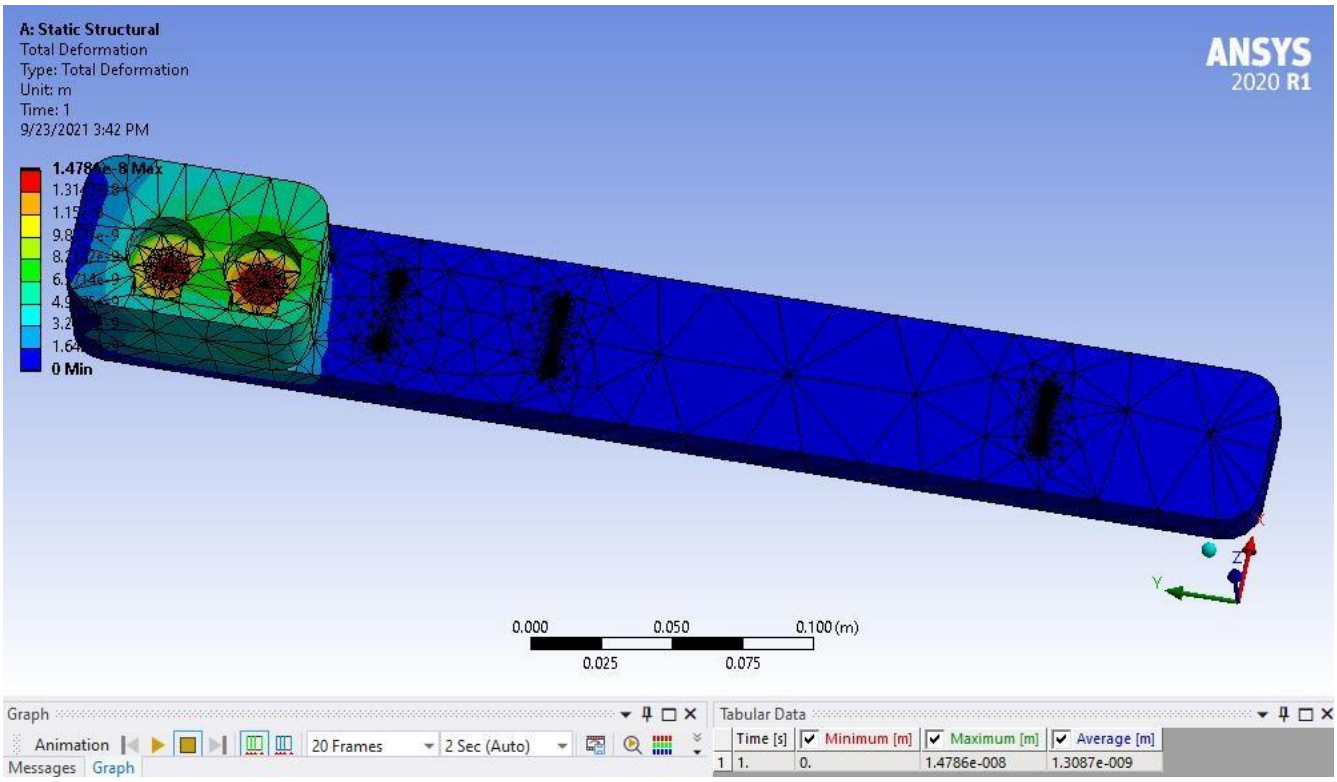


Figure 34: Total deformation of motor holder compartment

Average deformation was 1.3087×10^{-9}. The maximum deformation occurred at both the base and edges with a value of 1.478x10-8

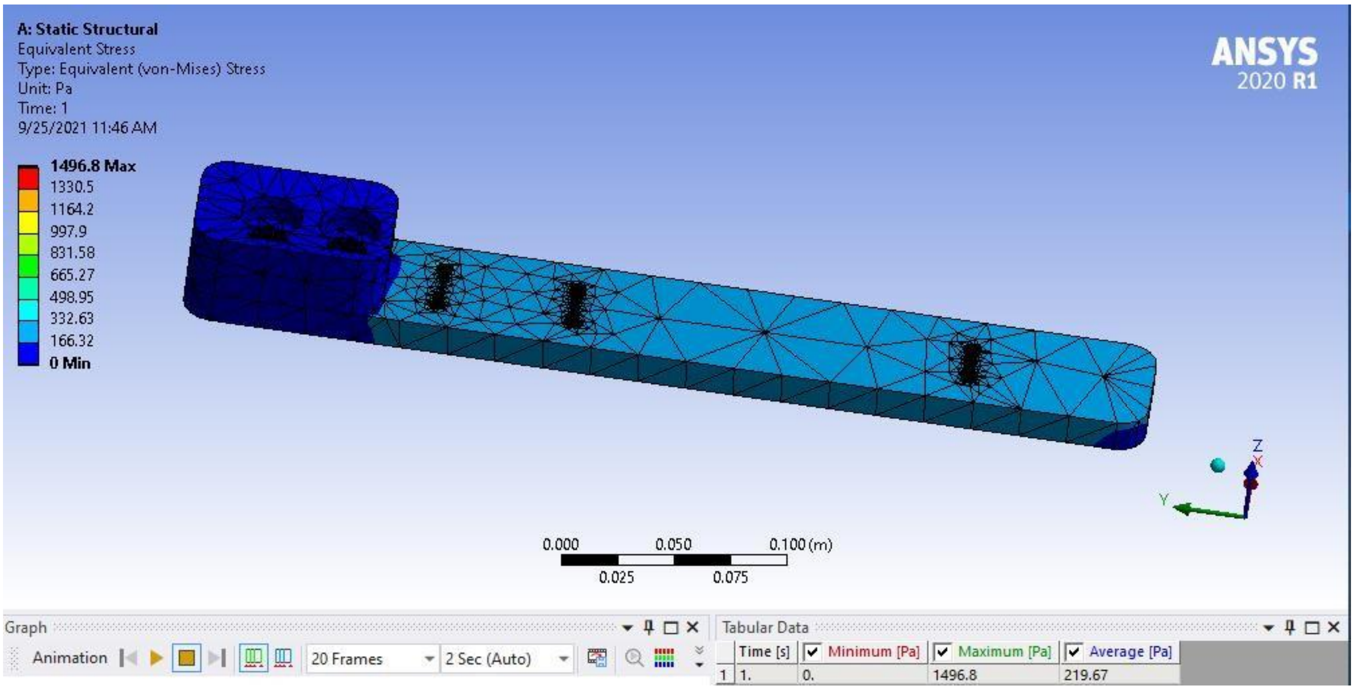


Figure 35: Equivalent (Von-mises) stress for base model

Figure 36 shows the stress distribution on the main base model. The maximum stress value was 1.49kPa while the average stress was 219.67Pa.

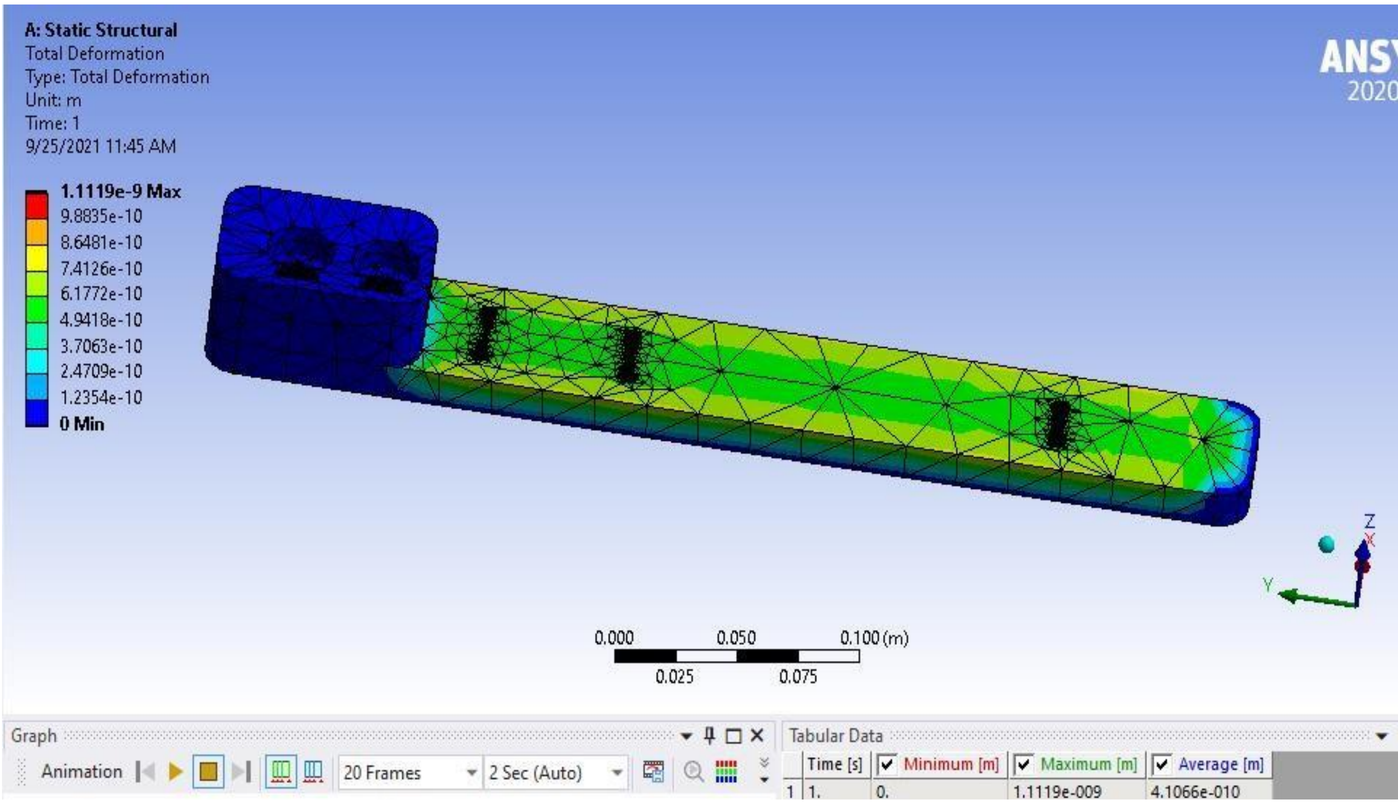


Figure 36: Stress distribution on the main base model

The maximum deformation occurred at the edges of the model as shown in Figure 36. Average deformation was 4.107×10^{-10} while maximum deformation was 1.11×10^{-9}

5.4 ANALYSIS OF THREADED ROD

In our design, the threaded rod is responsible for converting the rotatory motion of the bipolar stepper motor into a linear motion. To do this, the rod is coupled with the motor using a coupler. The nut moves along the rod when it rotates.

The following are the specifications of the material used for the analysis of the threaded rod:

Engineering Data: Material View

Structural Steel

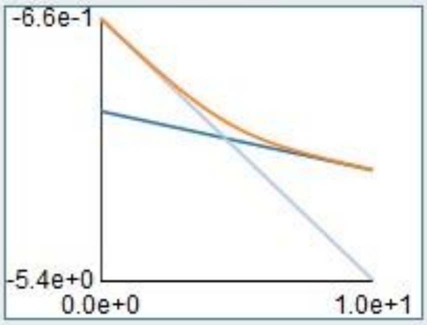
Density	7850 kg/m³

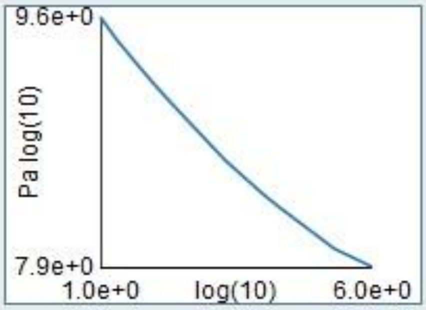
Structural

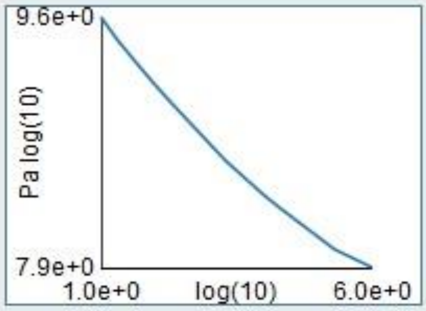
Isotropic Elasticity

Derive from	Young's Modulus and Poisson's Ratio

Figure 37: Total deformation of main base model

Young's Modulus	2e+11 Pa
Poisson's Ratio	0.3
Bulk Modulus	1.6667e+11 Pa
Shear Modulus	7.6923e+10 Pa
Isotropic Secant Coefficient of Thermal Expansion	1.2e-05 1/°C
Compressive Ultimate Strength	0 Pa
Compressive Yield Strength	2.5e+08 Pa
Strain-Life Parameters	

S-N Curve	
Tensile Ultimate Strength	4.6e+08 Pa
Tensile Yield Strength	2.5e+08 Pa

S-N Curve	
Tensile Ultimate Strength	4.6e+08 Pa
Tensile Yield Strength	2.5e+08 Pa

Thermal	
Isotropic Thermal Conductivity	60.5 W/m·°C
Specific Heat Constant Pressure	434 J/kg·°C

Electric	
Isotropic Resistivity	1.7e-07 ohm·m

Magnetic	
Isotropic Relative Permeability	10000

Geometry | Engineering Data: Material View

Figure 38: Specification for threaded rod

Because of the range of motion of the rod, the parameters to look out for during the analysis were the friction between the nut and the threaded rod as well as the moment generated when a torque is applied. The subsequent pictures explain each step of the process. Figure 39 shows the rigid supports used on the threaded before the application of the torque.

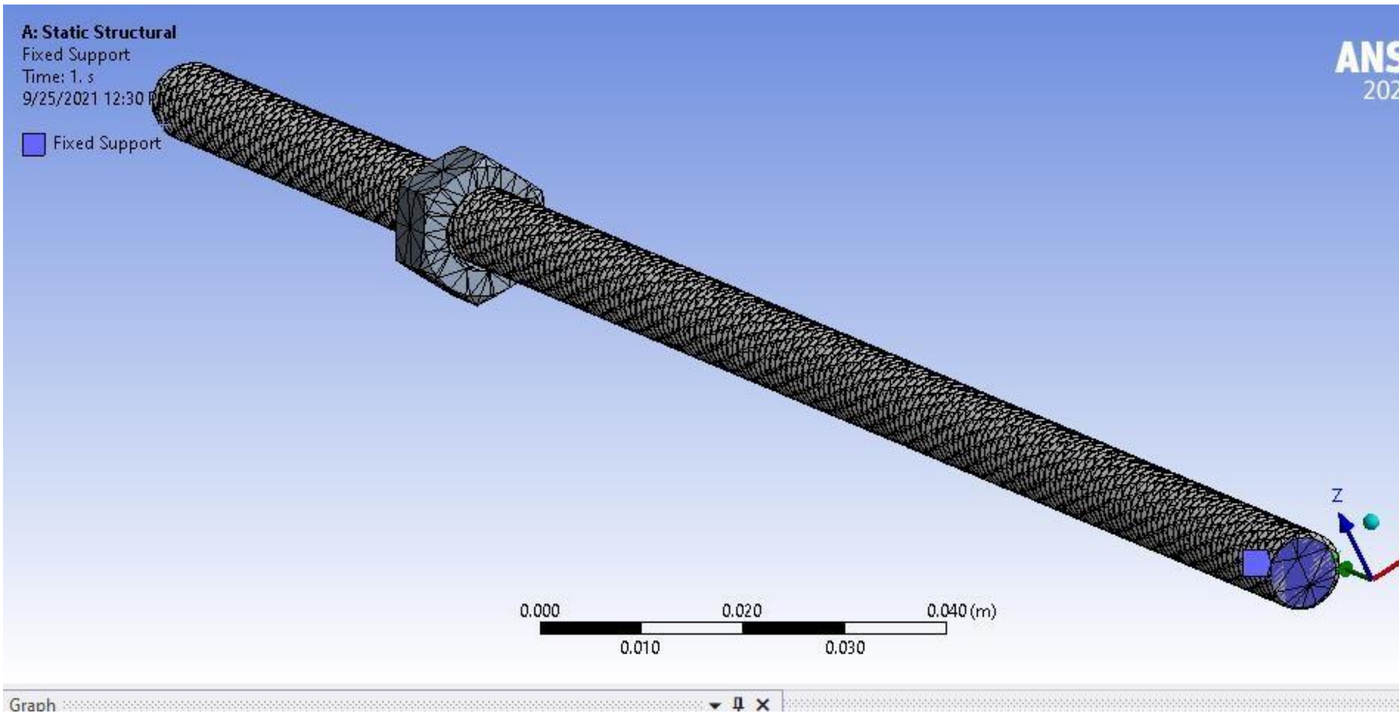


Figure 39: Application of a rigid support to the rod

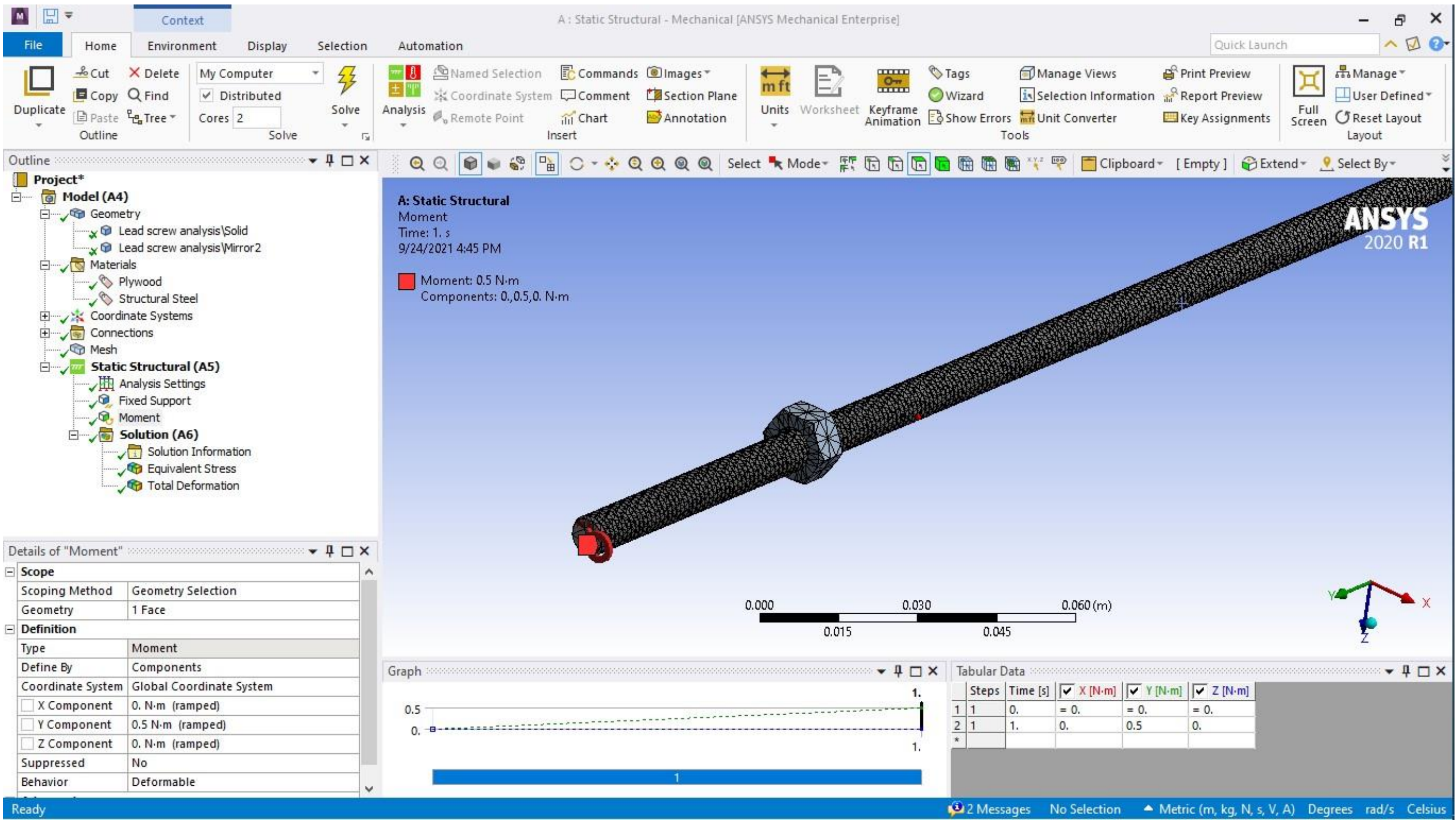


Figure 40: Application of torque to threaded rod

Figure 40 shows the application of a 0.5N torque at the opposite end of the rigid body.

Figure 41 shows the friction contact with 0.2 coefficient of friction

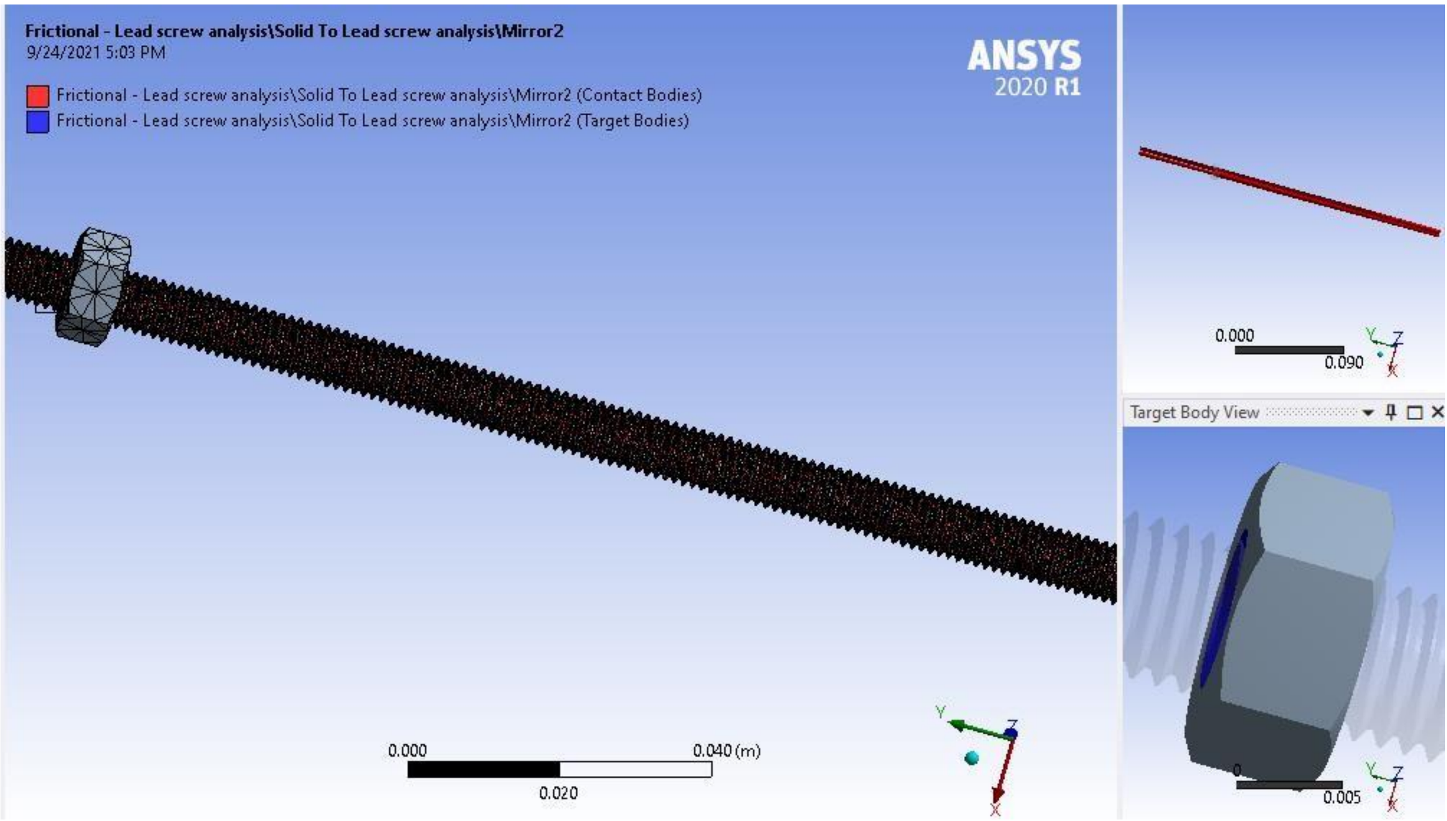


Figure 41: Friction contact

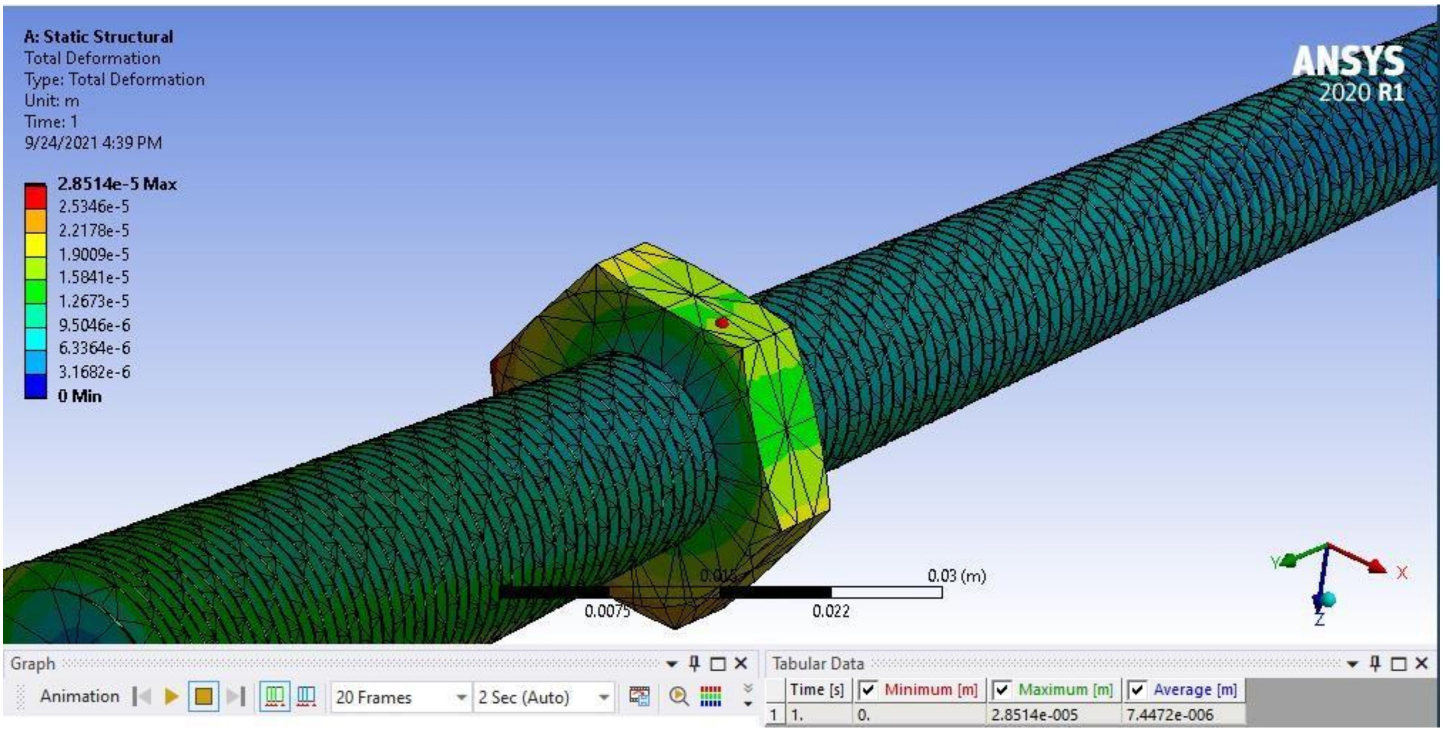


Figure 42: Total deformation of lead screw and nut

The nut had a total deformation of 2.85×10^{-5} occuring at the edge point of fixation as shown in Figure 42.

DISCUSSION

5.6 DICUSSION ON SIMULATIONS

The main variable in this discussion is the analysis of the stresses generated on the base model, threaded rod and nut. The various regions are analysed to identify the parameters that affect them.

5.6.1 Base model

The main base, where the bipolar stepper motor, lead screw and 3D printed materials rested is analysed as follows:

The main surface recorded good results with respect to the stress concentrations. The average stress generated was 219.67Pa while the maximum stress recorded was 1496.8Pa. Resulting in a total of 4.107×10^{-9} as the average deformation while 1.11×10^{-9} was the total deformation. The deformation resulting from these stress values are negligible. From this analysis, it can be concluded that the base model is capable of executing its task of holding the various components.

The unipolar motor compartment recorded low stresses. This is because compared to the other forces exerted on the base, the motors are light weight and therefore cause quite an insignificant deformation. The maximum stress generated on that part of the base was 30.45kPa resulting in a deformation of 1.478×10^{-8}. It was noted that a majority of the stress generated was concentrated in the holding compartments (holes). Moving further away from there, less stresses were generated. This is important as it shows that material choice for the project was good.

5.6.2 Threaded rod

From our simulations, the threaded rod recorded very large stress values. This was quite understandable, considering the work it does in converting rotatory motion into linear motion. Maximum stress was recorded at the point of application of the torque, which is the moment generated by the stepper motor. The stress value decreases as we move along the rod further away from the point of application. Average deformation recorded was 7.44×10^{-6}. In conclusion, the threaded rod will function well over a good period of time.

5.4 DISCUSSION ON BUILT MODEL

The current model is an upgrade of an already-existing model (ANET 1.0). The previous model had its positives and negatives, our task was to enhance the positive aspects of the model while working on all the problems inherited. The main issues with the device are discussed below, together with the techniques used in tackling them and the final results.

WOBBLING EFFECTS

Wobbling effects of the threaded rod. In the recommendation spelt out in the ANET 1.0 report, the threaded rod was indicated as the cause of the wobbling or inefficient movement of the mechanical components. A lead screw was recommended to solve the problem so we placed an order for one. However, after testing the device for multiple times, we found out that the wobbling was a combined effect of poor coupling and non-linear guiding of the pusher block. To test our hypothesis, a new coupler was purchased to serve as the interface between the bipolar stepper motor and the threaded rod. The guiding rod was also replaced with a new one that is exactly 8mm to firmly restrict the pusher block and force it to move only linearly. The result of this test shows impressive improvement in the movement of the mechanical components altogether. We had problems with arrival of our lead screw and some other components which we believe will make the device works even better but comparatively, ANET 2.0 produces nearly no wobbling effects.

OVERHEATING

Another challenge the previous model had was with overheating of the stepper motor and unnecessary noise during motion. Per the discussions and recommendations on the previous model, the problem associated with the choice of motor driver for the bipolar stepper motor. This discussion was tested and proven true. To resolve the problem, an Arduino motor shield was purchased and interfaced with the Arduino mega. From our tests, we found out that the motor driver wasn't the only requirements for smooth movements of the motor but power requirements were equally important. The motor needed two separate power sources to work. One from the Arduino Mega microcontroller to generate pulses and a separate 12v external power supply from the driver to the motor. When we tapped the 6v output of our first transformer as output for the motor, there were rigorous vibrations and a lot of noise generated as a result of power lag. This was resolved by connecting the output 6v to another transformer and stepping that transformer up to 12v, enough to power the motor perfectly with little or no noise and unnecessary vibration.

AIR BUBBLES

To solve the issue of air bubbles being trapped in the device, we decided to replace the 3Dprinted valves with airtight off-the-shelf valves. That way, blood components only come into contact with sealed blood-compatible components. Shafts were rather 3D printed to interface the valves with the motors. The only problem we encountered with the new valves was with producing enough torque to turn them. The valves were designed to be turned manually by hands and therefore overburden the unipolar stepper motors we employed. To solve this problem either one of these options will have to be employed: • Replacing the motors with higher torque motors

•Designing new valves specifically to fit our design

One of our objectives is to design a low-cost device and therefore changing the motors defeats that purpose since it will result in an entirely new design. The motors are also expensive. The more efficient option is to locally manufacture valves 3-way valves that will interface our motors perfectly.

Due to the delay of a number of our ordered products, we had to compromise on building an entirely new model as planned from the beginning of the project. Rather, we merged components from the previous mode and integrated the old design with our new design. Our initial design was different from the current one we are implementing. Our design involved a stationary lead screw and a moving non-captive stepper motor. Because the new motor was not delivered as scheduled, we were forced to quickly revisit the existing design and rather than change it entirely, improve it into a more efficient model.

The mechanical components function better with efficient electronics and sound programming. A lot of effort was put into these two aspects of the project to ensure that ANET 2.0 works very efficiently. The result of all the various sections of the project coming together, is an efficient device, capable of running more than 30 exchange transfusion cycles with minimal heating; little to no noise; smooth transitioning and an overall better coordination.

5.5 COST ANALYSIS

One of our specific objectives in carrying out this project was to design a low-cost device. To achieve this, we had to employ the use of affordable but quality products. Also, some components were locally made to cut cost. The cost analysis of the project is detailed below:

Table 7: Cost Analysis

Component	Quantity	Budget (GH₵)	Actual (GH₵)
Arduino Mega 2560 R3	1	100	100
Non-captive bipolar stepper motor (NEMA 17)	1	750	160*
28BYJ-48 Unipolar stepper motors	2	100	40

ULN2003 Stepper Motor Driver	2	100	20
L298NH Bridge Motor Controller	1	150	-
LCD & i2c	1	50	40
Touchpad	1	20	-
Jumper Wires	-	20	35
Guiding plate	1	200	20
Power source	1	40	70
Air-tight valves	2	100	40
3D Printed components	-	200	150
Base component	1	100	80
Arduino Motor shield	1	30	30
Electronics housing	1	50	45
Voltage booster	2	50	50
Shipping expenses	-	300	250
Miscellaneous		300	250
Total		**GH₵2660**	**GH₵1380**

Comparing the cost incurred in designing this device to the cost of similar medical devices, out device is far cheaper. Also, mass production will lead to an even cheaper cost.

CONCLUSION

From the beginning of the project our main objectives were to design and build a low-cost automated exchange transfusion device. We had to opportunity to continue progress from a previous model which had a number of problems ranging from: vigorous shaking of the motor when in motion; heating and unnecessary noise. Each problem was tackled either from the perspective of programming, electronics or mechanics. The individual sections were then integrated into what is now called ANET 2.0. The device is cost-effective, works efficiently and most importantly user friendly. The entire device can be powered with as little as 6V dc batteries because of the innovative power transformation system used.

RECOMMENDATION

We recommend the local design and production of blood-compatible 3-way valves specifically to interface our design. This should eliminate the current complications that come with using off-the-shelf valves.

APPENDIX

Appendix 1: Full Arduino program for running the device is available on GitHub at

GitHub - Automated Neonatal Exchange Transfusion Code

References

Alkén J, H. S. (2019). Rates of Extreme Neonatal Hyperbilirubinemia and Kernicterus in Children and Adherence to National Guidelines for Screening, Diagnosis, and Treatment in Sweden. *JAMA Netw Open*.

Brian Berman, A. K. (1979). A new method for calculating volumes of blood required for partial exchange transfusion. *The Journal of Pediatrics*.

Cogan, D. G. (1978). The Rise and Fall of Eponyms. *Arch. Ophthalmol*.

Corinna Underwood, D. S. (2019). Exchange Transfusion. *Healthline*, 2-3.

Elizabeth, J. (1975). Phototherapy in neonatal hyperbilirubinemia. *Aust Paediat. J*.

Falciglia, H. (2016). Past and present in neonatal exchange transfusion. *Archivos Argentinos de Pediatria*.

Giancarlo Liumbruno, F. B. (2009). Recommendations for the transfusion of red blood cells. *PubMed Central*.

Gluzman, N. (2019). Everything You Need to Know About Jaundice in Newborns. *eMediHealth*.

Kalakonda, A., Jenkins, B. A., & John., S. (2021). Physiology, Bilirubin. *National Centre for Biotechnology Information*.

Kaneshiro, N. K. (2021). Exchange Transfusion. *MedlinePlus - National Library of Medicine*, 1-2.

L M Gartner, K. S. (1977). Development of bilirubin transport and metabolism in the newborn rhesus monkey. *National Library of Medicine*.

LK, D. (1947). Erythroblastosis foetalis or haemolytic disease of the newborn. *National Library of Medicine*.

Luke A. Jardine, P. W. (2011). Clinical Evidence of Neonatal Jaundice. *National Library of Medice*, 3-6.

M Funato, H. T. (1997). Trends in neonatal exchange transfusions at Yodogawa Christian Hospital. *Acta Paediatr Jpn*.

Nader, E. S.-J.-D. (2019). Blood rheology: Key parameters, impact on blood flow, role in sickle cell disease. *Frontiers in Physiology* .

Safer Care Victoria. (2018). Umbilical vein catheterisation for neonates. *Neonatal ehandbook*.

Teichler-Zallen D, C. D. (2004). The Rhesus factor and disease prevention. *History of Medicine*.