Good Code Preachers

*Note: This is a documentation of the project named RESCUE ROBOT

Aditya Kumar
dept. of Electronic Engineering
Hochschule Hamm Lippstadt
Lippstadt, Germany
aditya.kumar@stud.hshl.de

Saikot Das Joy
dept. of Electronic Engineering
Hochschule Hamm Lippstadt
Hamm, Germany
saikot-das.joy@stud.hshl.de

Manoj Luitel
dept. of Electronic Engineering
Hochschule Hamm-Lippstadt
Hamm, Germany
manoj.luitel@stud.hshl.de

Muhammad Moaz Amin
dept. of Electronic Engineering
Hochschule Hamm Lippstadt
Hamm, Germany
muhammad-moaz.amin@stud.hshl.de

Evrard Leuteu Feukeu
dept. of Electronic Engineering
Hochschule Hamm-Lippstadt
Hamm, Germany
evrard.leuteu-feukeu@stud.hshl.de

Patrick Stephen Eteng
dept. of Electronic Engineering
Hochschule Hamm Lippstadt
Hamm, Germany
stephen-eteng.patrick@stud.hshl.de

***Abstract*—The rapid developments and innovations in autonomous and artificial intelligence systems have made a big impact in our lives. This has led to a reduced dependency of human beings in high risk fields like rescue operations. As these systems can be more precise and reliable, they are more efficient and effective and come without the risk of lives. In this paper we have described our prototype for rescue missions on land and water, where the primary focus is on water rescue operations. It also discusses the requirements, model designs and the algorithm for path detection of the robot.**

I. Introduction

Despite the huge technological upgrades over the years, disaster and accidents still remain a frequent issue in several parts of the world nowadays. These disasters either occur naturally like earth quarks or are caused by men e.g. fire incidents. In most cases, disasters demand massive rescue missions. Rescue missions either by land, water or air require a lot of preparations and precautions in order to avoid accidents not only for victims but also the rescue team. Traditionally most rescue missions involve teams of humans, who need to operate in very hostile and very dangerous sites. Keeping this in mind, the use of rescue robots and artificial intelligent systems has become a necessity. However, the fact that rescue robots have to operate in risky and dangerous environments implies the need for such machines to be very precise with the task assigned. These robots are designed in such a way that they could operate almost autonomously or with minimal human control. As our project we were tasked to research, design and develop a rescue robot that could work on either land and water.

II. Requirements Specifications

A. UML Diagrams

1) Activity Diagram: A rescue robot is an unmanned vehicle which is designed for the purpose of aiding in the search and rescue of humans or animals (depends on what is to be rescued), whom happens to find themselves in a disastrous situation. The purpose of this paper is to design a robot that is capable of maneuvering obstacles, search load and also perform rescue operations accurately. The activity diagram in Figure 1 below is used to show the behaviour of our system as well as model the activity flow of our rescue robot, as it interacts with various environments i.e., land and water scenario. Firstly, a task is assigned to an actor (user) to commence the system operations. When on land or in water the Robot has the ability to maneuver over an obstacle. The system receives rescue information, and the robot determines the best route to take. When it arrives the rescue area, it searches for load to be rescued. The search continues until the load is found and thereafter takes it to a secure location.

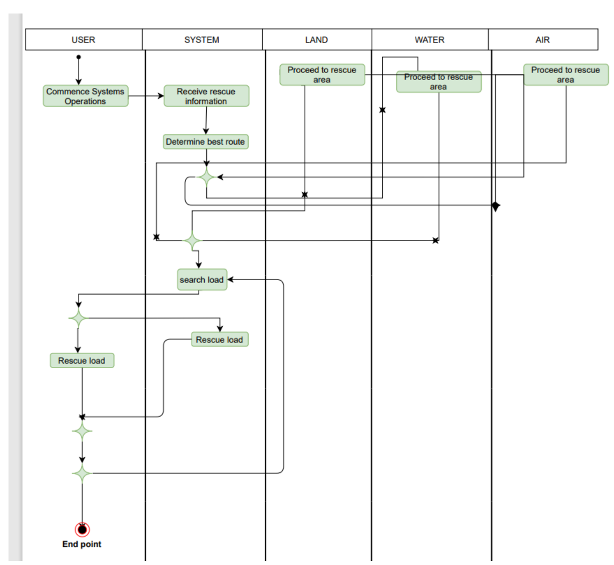

Fig. 1. Activity Diagram

2) Use Case Diagram: Since , Our developed rescue robot prototype has robotic arms, It can do various tasks , that means it is not limited to some specific tasks. In the use case diagram, only a few number of use cases are mentioned , but our robot is not strictly bounded to these only. Some major uses of this rescue robot is:

1) It can reach to the Target following the shortest possible path.
2) It can move in water as well as in land.
3) It works automatically and it activates itself automatically when it gets the accident notifications, No driver or operator is needed.

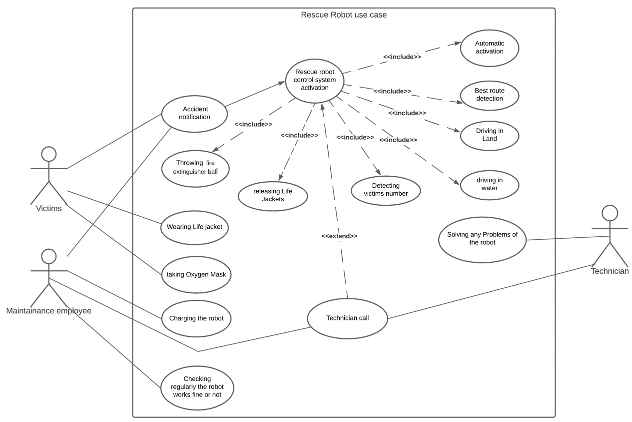

Fig. 2. Use Case Diagram

B. SysML Diagrams

1) Requirement Diagrams: Since the beginning, we had more ideas or scenarios with which the rescue task could be performed. The concept of environment/scenario was concertized, and the requirement was specified in a way that the system can drive over some sort of obstacles (with some limitations), heading smoothly on land and on water to the rescue spot as well as returning to the initial position. To come up to the final specification the team has gone through elicitation, analysis, specification and validation steps.

Requirement diagram (SysML diagram) is used to model the relationships between requirements (movement. Performance, interface etc.), and model tracibility relationships between requirements and items in the model. The diagram represented below is the initial SysML requirement diagram.

2) Constraint Diagram: Constraint Diagram generally encompasses all the Blocks or hurdles that can cause our complete Robot to fail or function less effectively. Our Initial approach regarding this had certain major constraints which could lead to failure of the complete rescue operation for example if the Robot is faced with a hurdle in front, it won't be able to move past it which would lead to a disastrous situation.

Other constraints that we looked into included Cost which made us think of a strategy in which we can make our robot very cost effective so it is accessible for all.

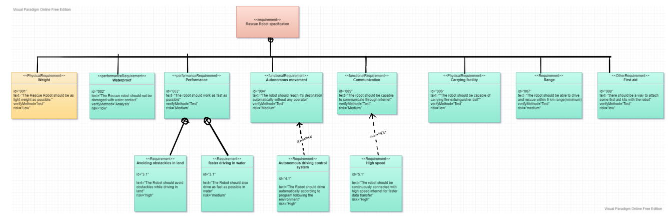

Fig. 3. Requirements Diagram

Another important constraint which we included is the mobility on both land and water. Mobility on both the surfaces involves around a lot of other factors as well which include the total weight of the robot plus the material it is made of, it's air resistance equation and etc. When designing the final prototype we did have to look into all of the constraints present in the constraint diagram and come up with such a solution which made our robot functional. There are certain constraints like fuel efficiency and power that can be taken care of in the near future because altering them would result in enhancement of body structure as well. Figure 4 represents Constraint Diagram of the Project.

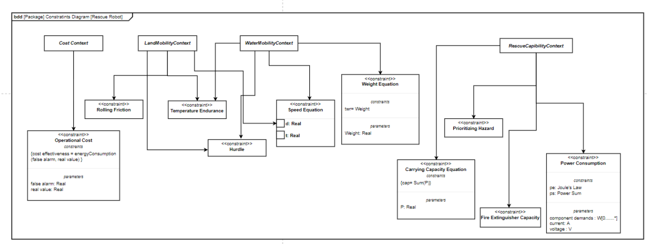

Fig. 4. Constraint Diagram

3) Block Diagram: A Block is a modular unit of system that encapsulates its contents, which include attributes, operations and constraints. Blocks can be connected to other Blocks to form composite structures, and can be decomposed into parts to expose internal structures [from the lecture - SystemsEngineering 03]. Our block diagram represents the initial system and subsystems interaction / relations of our rescue robot prototype. We have defined various actors on the left side responsible for the control and maintenance of the robot. On the right side we have an external block that takes account of the environment that is not in control of the system such as weather, external objects and encountered path type which is either land or water. Further we have also defined the Subsystems such as-

- power
- Breakdown
- Interior
- Exterior

Interior Subsystems include Health Monitor and Emergency Control system. The Exterior Subsystem consists of Subsystems such as sensors and lighting. Keeping in mind that these were the initial System architecture that the team thought off. All of these <<blocks>> are connected with relations like composition, association and specification.

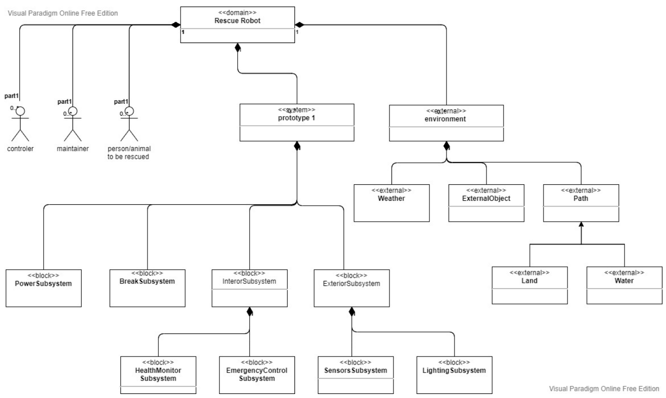

Fig. 5. Block Diagram

4) Block Definition Diagram: Figure. 6 represents the subsystems of rescue robot system. Identified subsystems are environment, body, Power, Navigation and control, movement regulation, data handling, Feedback encoding and stakeholder. They are further associated with number of blocks to make up a subsystem.

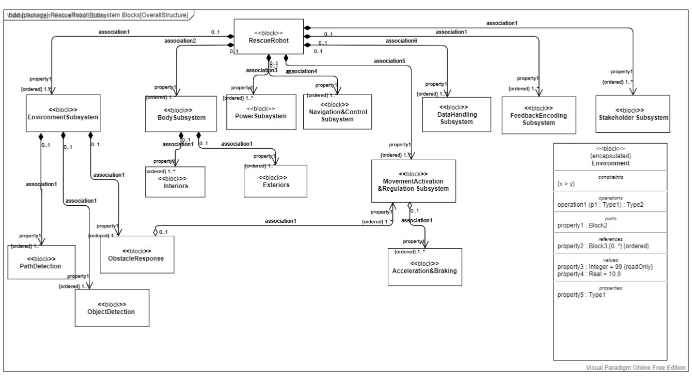

Fig. 6. Block definition Diagram for Subsystems

Hierarchical Control Pattern: The system follows an architecture model of Hierarchical Control Pattern that uses two types of interfaces: control interfaces that monitor and control the behaviors are achieved and functional interfaces, which provide the services controlled by the other set of interfaces.

The control interfaces have set the quality of service, such as fidelity, accuracy, and so on, as well as selected policies that govern how the execution proceeds. The functional interfaces have executed the desired behavior using the quality of service and policies set via the control interface. This pattern has allowed complex systems to be viewed at many levels of abstraction. In addition to that the use of separate control and functional interfaces provides a simple, scalable approach that can be used when configuring. This is implemented as one peer and-state per elements in the state chart of the controller, which is very useful, particularly when the states of the various subordinate parts are not completely independent. The use of and-states makes it much easier to ensure that the subordinate parts are always in mutually compatible states. When the control or configuration behavior of the subordinate parts is not modal, then other means is employed to track the behavior of the subordinate parts.

III. Development of Prototype

A. Components of Robot

1) Jet-Ski Propeller:

- Figure 7 shows a Jet Ski impeller (five fan) blade which is a component of the Jet Ski engine. An impeller blade is the rotating part of the pump that is used for the increment of the water pressure. The impeller works in such a way that the pump sucks water through the intake grate, and pushes it out from the steering nozzle.

Fig. 7. Impeller Blade

- The rudder is located at the stern behind the impeller, and it is used for steering the boat and maneuvering obstacles along its path.

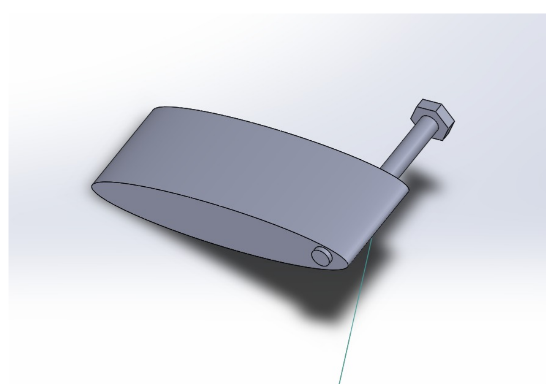

Fig. 8. Rudder

- Figure 9 displays how the robot jet ski works. Firstly, when driving in a waterlogged area, large volume of water gets into the inlet and outlet of the robot Jet, thereby drawing the water into the propulsion channel, then forcefully ejects it through the spherical bottom shape into the nozzle of the Jet. This process transforms the energy flow into rotational energy that rotates the blade and enables the robot to move on water surface.

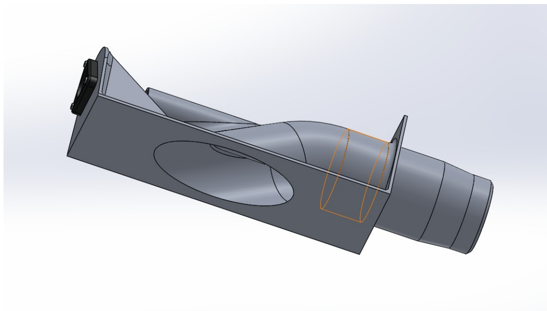

Fig. 9. Frame of Propeller

- This Robot Jet Ski Propeller shows the velocity and turbulence intensity from the initial plane of our Robot. The characteristics flow rate of a boat jet propeller was studied in-order to ascertain and understand its mode of operation. We were able to understand that the thrust of the propeller enables the robot to move on water and without it nothing happens. Just like tires of a vehicle, our jet ski propeller is the primary connection between the robot's engine and water, which is responsible for forward and backward movement while in water. We also observed that a 5-blade propeller is preferable to a 4-blade because it suppresses vibration and enhances acceleration by inserting more blade area in water, which also brings about optimum speed and stability during turbulence.

For the hypothesis we tried to study and compare the properties, material and statistics of the propeller used in our rescue robot, with different impellers. Using a small experimental setup and assumption based on information retrieved form our rescue robot, we tried to determine the horsepower, power, and thrust force of the propeller considering the weight and dimension of the robot. Firstly, the propeller being the ultimate dependent variables differed by only their number of blades i.e., four blades and five blades. It is important to note that each blade across all propellers are the same geometry and area. The area of water contact across each propeller impacts the result of the final decision making. Prior to the experiment it was hypothesized that the five-blade propeller would be the most efficient of the propellers. Furthermore, much of the evidence gathered suggested that this hypothesis is presumed correct, more data and research are needed in a more stable experiment to confirm this claim as true. During this hypothesis, we noticed the formation of cavitation which is simply the formations of bubbles due to the rotating machinery of the blades hence reducing the efficiency of the propulsion due to less pressure of water on the blades.

Fig. 10. four and five fan blade

It was also hypothesized that out of a four blade, and five blade propeller the five-blade propeller would have the highest efficiency as compared to the four blades propeller as theorized from the above background information, we observed that all impeller blades began undergoing cavitation at roughly the same time based on our research. Moreover, the material also impacted the

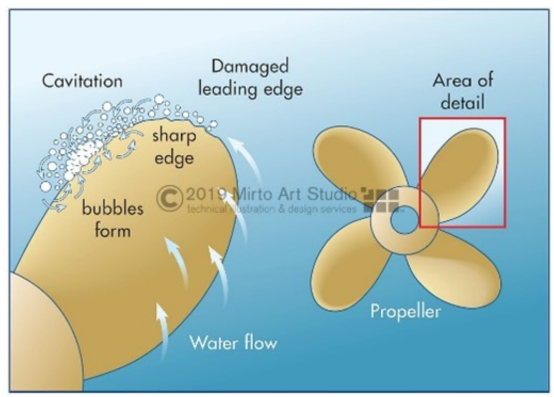

Fig. 11. Schematic

performance of the propulsion likewise plastic weighs less than metal, consequently it therefore takes less horsepower to rotate the impeller, making the engine more efficient. Moreover, due to the light weight, plastic impellers put less stress on the water pump bearing making it to last longer as compared to metallic. Assuming all propellers were supposedly printed from the same 3D printing machine. Overall, this hypothesis was a great learning experience that must be improved if its data is to be clearer and more accurate and practical for future testing.

After understanding the principle of operation of our jet ski 5-blade propeller, we discussed on the thrust which is a force that acts on the surface of a liquid as well as on the propeller blade. According to Newton's law which states that, to every action there is an equal and opposite reaction. It was concluded that the thrust that was generated by the propeller was due to the fluctuating water. The velocity of the circumference between the propeller blade and the fluid surface was obtained in the rotational direction of the propeller. The forward and backward movement of the robot was also discussed.

Jet ski conclusion: Considering the metal propeller, its structural characteristics are only affected by the material properties, the geometrical characteristics of the blade, and assembly method parts of the blade and the shaft. The geometrical characteristics are determined by the geometrical parameters of the blade, such as diameter, disk ratio, surface area, etc., and the connection type depending on if the design is the fixed pitch propeller (A propeller whose blade angle at the section under consideration is fixed and cannot be changed.) or the variable pitch propeller. However, we decided to go for a pitch propeller due to its efficiency and constant RPM. also, the structural characteristics of the plastic propeller are very different from those of the metal propeller. Based on the above analysis, plastic propeller is most appropriate on our rescue robot. Nevertheless, due to limited research and equipment's, more details in the design and analysis methods of plastic propeller are needed to provide a good opportunity to improve the performance of the propeller.

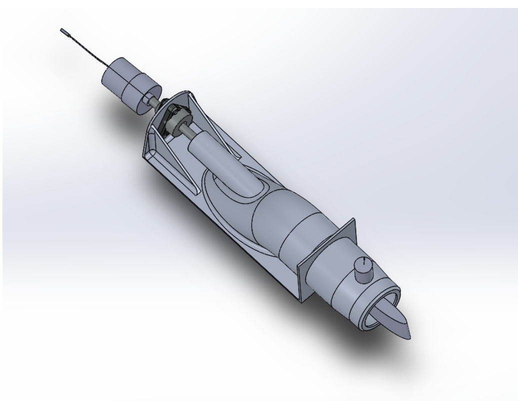

Fig. 12. Final Assembly of Propeller

2) Motor-Wheels: Software Tool: Solid Works2020 Motor-Wheel: In terms of energy and motion efficiency on flat surfaces, we have considered that wheeled robots are the most efficient. This is because an ideal rolling (but not slipping) wheel loses no energy. A wheel rolling at a given velocity needs no input to maintain its motion. This contrasts with legged robots which suffer an impact with the ground at heels trike and lose energy as a result. For simplicity our robot has four wheels or a number of continuous tracks. The wheels are surfaced with rubber materials with straight-lined strips to maintain sufficient friction on the path. We tried to create more complex wheeled robots with only two wheels as these can have certain advantages such as greater efficiency and reduced parts, as well as allowing a robot to navigate in confined places that a four-wheeled robot would not be able to. The rotation of front or rare two wheels at one time are controlled to change the direction.

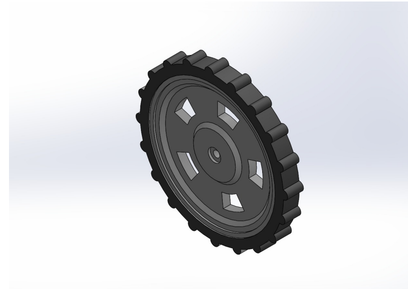

Fig. 13. Wheel Side View

The wheel system is designed, scaled and assembled on Solid works.The wheel is embedded through the rotating lever of the motor for the mechanical transference. We have used DC motor on each wheel to work well in our system as they allow the robot to be battery powered, which offers great advantages for a variety of robotic applications, particularly for our mobile and collaborative robot.Total of four wheel system are used for the movement and for changing the direction on the way. In the SW file, the rotating lever (size: 3mm) of motor is inserted to the wheel of outer diameter 5cm. The picture of final assembly is included here:

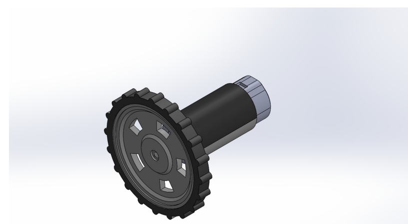

Fig. 14. Final Assembly of Wheel

3) Robotic Arm: Robotic Arm is one of the main components of the robot which is responsible for executing the rescue operation. There are a a lot of smaller Components involved in its composition.

- **Base** visible in Red colour is that component of the robotic arm which is going to attach to the main body of the robot and here we'll use screws to attach it. There are screw holes present on the outer radius for the attachment.
- **Shoulder or Hombro** is the Green part present in the robotic arm which makes 180° rotation possible for the whole arm. It is attached on top of the base. The Hombro is also providing a good height to the laterals which will be responsible for vertical operation.
- **Lateral** visible in blue Colour which are attached directly to the two lateral which allow angular movement to the arm so that it has ability to grab at the farthest possible location.
- **Rotor** in Blue Colour is the one component which is responsible for circular motion of the arm part.
- **Hand** visible in yellow is acting more like a palm. It comprises of certain Gears which rotate according to the

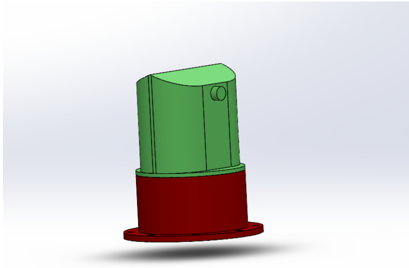

Fig. 15. Base of the Robotic Arm

required pressure or force needed to grab a person. It is the gears which drive the black levers to act as fingers.
- **Fingers** visible in black colour are the most important component of the Robotic arm which make the grab operation functional. For a better grip some rough surface is included in the design so that in general the grip is very tight so the rescued personnel do not let loose while movement through different surfaces.

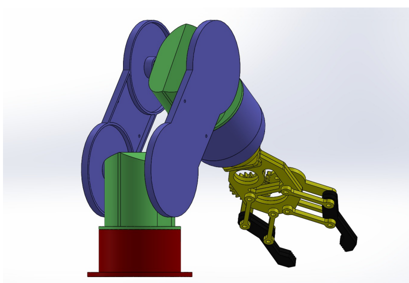

Fig. 16. Back View of Robotic Arm

Final Assembly of Robotic Arm

Figure 17 shows the complete assembly of the Robotic arm. The mechanism for operation that we used over here is Gears in the frontal part of the arm which we have created with Mate relation by using Mechanical Mate of Gear with .90mm ration. This made the mechanical operation possible as one gear rotates the other which in turn moves the complete arm. In side of the Arm we plan to use a FPGA Xilinx 3e which will simultaneously handle the rotation and then the mechanical operation as well.

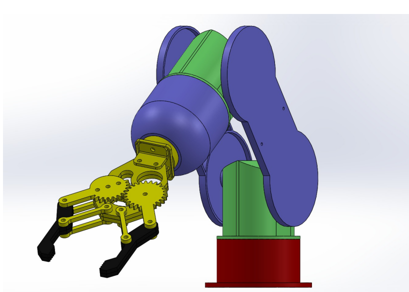

Fig. 17. Final Assembly of Robotic Arm

4) Chassis: The chassis of the robot was built in the end after the developments of all the other parts. This was necessary to ensure that the body was compatible with all the other parts. It is made in 3 sections: the bottom hull, the central frame and the lid.

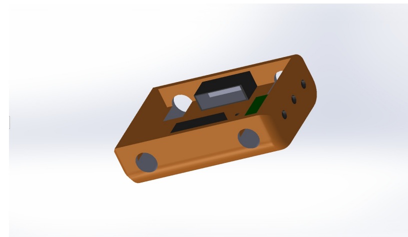

Fig. 18. Top / Side view

- **The hull** is the bottom most part and is also the part that would submerge in the water. The water jet is also placed in this part of the robot and is connected to the mainframe via bolts. The jet can be removed for repairs by removing the screws and sliding it out.
- **The central frame** is one of the most crucial sections of the robot. This section consists of the four motors for the wheels to operate on the land, the micro controller that analyzes all the operations, sensors such as cameras, proximity sensors etc crucial for the operation of the robot and of course the battery holders. The frame is open from top that helps in repairs and replacement.
- **The lid** not only acts as the cover for the robot but also the arms of the robot are positioned here. These arms are screwed and can be easily removed for repairs.

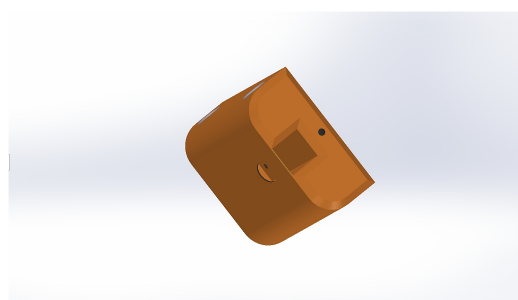

Fig. 19. Bottom / Back view

5) Complete Assembly: The final assembly is the compilation of all the parts of our Robot. The robot will be using MSP430 micro controller as the primary control unit to execute the tasks. It will also include light, proximity, sound, and temperature sensor. The powerhouse of the Robot that is the batter pack will be Lithium Polymer or LiPo/LiPoly Packs. We will also be using Silicone Cables to keep the power transmission water tight. The Robot will work well with synchronised parts, and would be able to preform rescue operations efficiently.

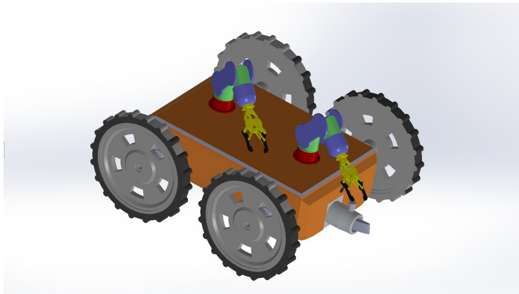

Fig. 20. Final Assembly of Propeller

B. Operations

Driving on Land :

- There are four motor wheels in the rescue robot to move over the land area. while it needs to turn towards the right direction, two motor attached to the left sided of the robot starts rotating forward and two motor of the left side start to rotate backward. And while it needs to turn towards the Left direction, two motor attached to the right side of the robot start rotating forward an two motor of the left side start to rotate backward. To go forward, all motor wheels rotate forward and backward rotation of four motor wheels for the movement towards backward movement.

Driving on Water:

- Bringing help to victims in flooded areas or bodies of water can quite be challenging because disasters such as floods and hurricanes are very unpredictive hence making the rescue efforts complicated. It's often hard for rescuers to determine the likelihood of finding survivors in the water. In cases where conditions make the water especially rough, they may not want to take chances until they know for sure that survivors need help. So this hesitation can be minimized thanks to our rescue robot. Equipped with batteries, motors, optical and navigational sensors, the Jet propeller permit the propulsion of the rescue robot and the robot's body system offers a light weight which can float on water and is able to navigate on water with the help of the rudder. The objective of Good code preacher's team is to create rescue system capable to move both on land and water. We aim to design and make a lighter, efficient and more cost-effective robot system as possible.

  Description
  - For the safety of the rescue team, the rescue robot can realize autonomous operations with the help of software. The rescue robot can transits automatically from land to water and can be deployed on water to go and save those drowning within 100 meters range. avoiding the risk and/or accident from work.
  - On water our rescue robot reaches people at risk through the help of cables where by the victims can hold on, for them to be pull over away from the disaster proximity.
  - Enclosed with safety body frame, the victims doesn't need to worry about their hair getting tangled by propeller. And, its rubber head can also prevent collision from hurting passenger.
  - The choice and the design of the propellers enable the Rescue Robot to be work effective and applicable to bad weather conditions.
  - Optional payload, such as speaker for talking to those at risk and nervous and for speaking while searching and life jacket can be deployed to help victims from drowning.
- The people who race to the scene and rescue survivors after disasters are undoubtedly brave as they accept the risks of putting themselves in danger to help others. Now our days, the use of technology plays a prominent role in making the circumstances less dangerous.

IV. Implementation of the Programming Task

A. Primary

This is the primary approach (wave front approach) for the programming tasks.
To reach the target
(1) A new integer array of the same size of the map (world[200]) is created based on the given map (world array). Details are in the table diagram:

#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	\n
#	~	t	~	*	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	\n
#	~	~	~	*	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	\n
#	*	*	*	*	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	\n
#	O	O	O	O	O	R	O	O	O	O	O	O	O	O	O	O	O	O	#	\n
#	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	\n
#	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	\n
#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	\n

Fig. 21. World[200]

Newly created integer array is going[200].

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	\n
-1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	\n

Fig. 22. Going[200]

- So, where there is wall(#) in the map, there is -1 kept in the newly built array , and where there is Target (T) or Target(t) in the given map, there is 1 kept in the newly built array. And 0 for all other things.

After this, we start putting numbers in the going[200] starting from the position where there is 1. We check the surrounding (north, east, south, west) either there is -1 or not. If there is

-1, We ignore it, if there is 0, we increase the value of the starting position which was 1 at the beginning, and newly kept numbers to the surrounding is 1+1=2 after the increase. then, we start checking the surrounding positions individually from the position of the newly kept numbers and increase value from that newly kept number to (+1) increased.

This is done until there is no 0 in the same position in going[200] as robot position in Map.

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	\n
-1	2	1	2	3	4	5	6	7	9	0	0	0	0	0	0	0	0	0	-1	\n
-1	3	2	3	4	5	6	7	9	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	4	3	4	5	6	7	9	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	5	4	5	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	6	5	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	7	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	\n

Fig. 23. After Numbering on Going[200]

For robot movement:

- Search in the map, on which position there our robot is.
- At the same position in new array (going[200]) , find out what number there is.
- In going[200], in the position bth (suppose, in bth position, there is robot in the map),we see what number is there(suppose number is a).Then we check the surroundings(north/east/west/south) , which one is the (a-1).
- Our robot will move to that direction where there is (a-1). If there is obstacles ($*$) in that direction in the map, It destroys that obstacles and go on . (In code, it returns a value to destroy the obstacles).
- If there is two (a-1) in two direction, it selects any direction randomly.
- If there is water in that position and present position is in land (pre='O') in the map it returns value 5 to toggle water mode and also changes pre=O to pre = $\backsim$. vice versa for present position in water and next position in land.

To return to the home. (1) A new integer array of the same size of the map (world[200]) is created based on the given map (world array):
Details are in the table diagram:

#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	\n
#	~	R	~	*	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	\n
#	~	~	~	*	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	\n
#	*	*	*	*	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	\n
#	O	O	O	O	O	X	O	O	O	O	O	O	O	O	O	O	O	O	#	\n
#	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	\n
#	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	\n
#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	\n

Fig. 24. World[200]

Newly created integer array is returning[200].

- So where there is wall(#)in the map, there is -1 kept in the newly built array , and where there is Home (X) in the given map, there is 1 kept in the newly built array. And 0 for all other things. This happens only when robot moves from home.

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	\n
-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	\n
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	\n

Fig. 25. Returning[200]

After this, we start putting numbers in the going[200] starting from the position where there is 1. We check the surrounding (north, east, south, west) either there is -1 or not. If there is -1, We ignore it, if there is 0, we increase the value of the starting position which was 1 at the beginning, and newly kept numbers to the surrounding is 1+1=2 after the increment.
then, we start checking the surrounding positions individually from the position of the newly kept numbers and increase value from that newly kept number to (+1) increased.
This is done until there is no 0 in the same position in returning[200] as robot position in Map.

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	\n
-1	0	8	7	6	5	4	5	6	7	8	0	0	0	0	0	0	0	0	-1	\n
-1	8	7	6	5	4	3	4	5	6	7	8	0	0	0	0	0	0	0	-1	\n
-1	7	6	5	4	3	2	3	4	5	6	7	8	0	0	0	0	0	0	-1	\n
-1	6	5	4	3	2	1	2	3	4	5	6	7	8	0	0	0	0	0	-1	\n
-1	7	6	5	4	3	2	3	4	5	6	7	8	0	0	0	0	0	0	-1	\n
-1	8	7	6	5	4	3	4	5	6	7	8	0	0	0	0	0	0	0	-1	\n
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	\n

Fig. 26. After Numbering on Returning[200]

For robot movement:

- Search in the map, on which position there our robot is.
- At the same position in new array (returning[200]) , find out what number there is.
- In returning[200], in the position bth (suppose, in bth position, there is robot in the map),we see what number is there(suppose number is a).Then we check the surroundings(north/east/west/south) , which one is the (a-1).
- Our robot will move to that direction where there is (a-1). If there is obstacles ($*$) in that direction in the map, It destroys that obstacles and go on . (In code, it returns a value to destroy the obstacles)
- If there is two (a-1) in two direction, it selects any direction randomly.
- If there is water in that position and present position is in land (pre='O') in the map it returns value 5 to toggle water mode and also changes pre=O to pre = $\backsim$. vice versa for present position in water and next position in land.

```
int z = -1;
int move(char* world, int map_id)
{
 if (z != map_id)
 {
 z = map_id;
.. .. .. .. .. .. .. //Further codes
 }
.. .. .. .. .. .. .. //Further codes
}
```

Listing 1. Sequential Execution of Different Maps

Robot movement overview:

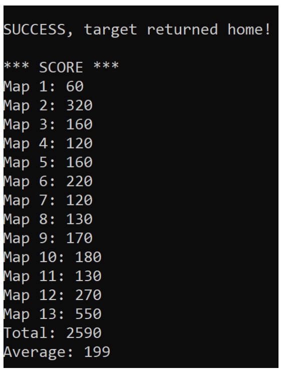

Fig. 27. Robot Movement Overview

B. Secondary

This is the secondary approach for the programming tasks.

As discussed before with the primary algorithm we were tasked to design an algorithm that guides the robot in different worlds designed by the professor. The code is created under the function int move(char* world, int map_id). In this paper we will be discussing only the last task as it covers the previous tasks as well. The following points describes the algorithm

- First we run a for loop that detects certain characters such as 'R' , 'T' , 'X', and 't' and assigns them to their integer variable index. The character represent -
  - R - The position of the robot
  - T - The target on land
  - t - The target on water
  - X - The return base
- Once the target is identified we divide our 1D world into xy coordinate system, that is the vertical line in the world becomes the y axis and the horizontal line becomes the x axis. We then divide the map in four quadrants keeping the Target index at the center of the quadrant. By doing this we assign x and y coordinates to the Robot index and the Target index, making us aware of their positions and helping us in the path detection.
- Next we get the character values of the next possible step in north, south, east, and west and also we get the character values diagonal to the Robot index.

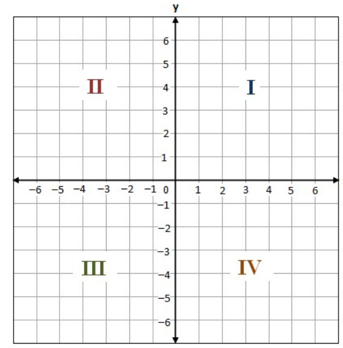

Fig. 28. Quadrant system

- Once we have all this information, we run a series of if statements that check through the conditions and assign the next step. This condition check is designed in layers
  - First we check if the driving mode is in land or water. This is a very important step because if we are in land driving mode but we are in the water region, the system would fail to operate.
  - If we are in land driving mode the following steps are followed-
    1) We check if the next step is land represented by 'O' on the map or if its water ' '.
    2) If the next step is 'O', we check in which quadrant the robot index is placed. By doing this we decide the path the robot should take. For example if the robot is in the 3rd quadrant it would go in the north direction.
    3) If the next step is ' ', then we toggle the mode to water driving mode first and then follow step 2.
    4) If we encounter and walls '#' we have conditions defined to reroute the path.
    5) If we encounter an obstacle '$*$' We remove that obstacle first and then we proceed further.
  - If we are in water mode and the next step is land, we first toggle to land mode and follow the steps mentioned above.
  - Once we reach the Target index we switch the center or the quadrant to the Base index, and then follow the same steps mentioned above to return to the Base.
- The quadrant system considering Robot index and Target index :
  - First quadrant : the conditions for this quadrant are:
    - $*$ Robot index [x] $>$ Target index [x]
    - $*$ Robot index [y] $<$ Target index [y]
  - Second quadrant : the conditions for this quadrant are:
    - $*$ Robot index [x] $<$ Target index [x]
    - $*$ Robot index [y] $<$ Target index [y]
  - Third quadrant : the conditions for this quadrant are:
    - $*$ Robot index [x] $<$ Target index [x]

- Robot index [y] > Target index [y]
  - Fourth quadrant : the conditions for this quadrant are:
    - Robot index [x] > Target index [x]
    - Robot index [y] > Target index [y]
- Though this algorithm works well in most of the world. It still has an efficiency rate of 99% as it fails to return to the base in the last world.

```
*** SCORE ***
Map 1: 60
Map 2: 320
Map 3: 160
Map 4: 120
Map 5: 200
Map 6: 220
Map 7: 120
Map 8: 260
Map 9: 170
Map 10: 150
Map 11: 130
Map 12: 340
Map 12: FAIL
Total: 2250
Average: 187 / 1 failures
```

Fig. 29. Results of task 4 / Algorithm 2

V. Limitations

One of the downsides of our rescue robot is that it requires constant electricity to run which is not climate friendly. Its manoeuvrability capability is limited to larger obstacles encountered along its path and it also have a limited number of loads it can rescue. It does not have any personal requirement; it needs to be programmed in order to perform a given task. Our robot has 75 percent dominance on water in comparison to land which is only 25 percent. These are just the few restrictions of our rescue robot.

VI. Evaluation

From the development point of view, the final results have been very satisfying. Despite encountering problems with the development of our prototype design, the overall project was a success. Regarding the design, we could not really develop the design the way we wanted to as the team lagged knowledge in Solid Works. In terms of programming and the initial research of requirements, the tasks were handled efficiently by the team. In The end our robot was reaching the target and returning back to the base efficiently and behaving in the correct manner doing what it was supposed to do at every moment.

VII. Closing Remark

As the main purpose of the whole project was to learn and develop the concepts that include the interactions between the physical world and the electrical world, the main concepts discussed in the paper are useful to understand how these interactions occur and how to deal with them. The underlying efforts were an essential door openers to the world of System development and prototyping, as it was a prime introduction to fields such as creating requirements document, block diagram, designing hardware, interaction design, project development,and so forth.

VIII. Division of Work

- Throughout all of the project, all team members sat together every week and discussed the tasks and divided the tasks among all team members and planned another meeting to see everything is going well or not.

A. Phase 1

The team was asked to work on the requirements documentation for the prototype that included UML and SysML diagrams that included activity, use case, constraint, block, block definition and requirements diagram, the team split each diagram with individual team members.

B. Phase 2

The team was asked to work towards the development of the design for our prototype. This was a very crucial phase for the team as we lagged the experience in designing on the system. We all had different design platforms but we decided to work on solid works. The team had to go through multitudes of YouTube videos to understand the concepts but we were able to design our prototype in the end.

C. Phase 3

The team was tasked to create the move algorithm for the robot in the programming tasks. After a lot of brainstorming we decided to split the team in 2 parts as we had two different approaches to follow to create the algorithm. This was the most efficient phase of the project and the team was very confident with the work.

Team Memebers

1) **Aditya Kumar:** Block diagram, The chassis, Final Assembly, Programming of Secondary Approach.
2) **Saikot Das Joy :** Requirements Diagram, Requirements specification document, Motor-wheel,Use case diagram Final Assembly., Primary Programming Approach(wavefront approach).
3) **Manoj Luitel :** Block Definition diagram(Subsystem refinement),Requirements specification Document, Motor-Wheel Design Scaling, Programming of Primary Approach.
4) **Muhammad Moaz Amin:** Constraint Diagram, Robotic Arm, Programming of Secondary Approach,Final Assembly.
5) **Evrard Leuteu Feukeu :**Use Case Diagram, Designing Jet Ski, Primary Programming Approach.
6) **Patrick Stephen Eteng :**Activity Diagram, Designing of Jet Ski, Final Assembly, Programming of Secondary Approach.

References

[1] Norman, Donald A. The Design of Everyday Things. New York: Doubleday, 1990. Print. https://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport.

[2] Zhmud, Vadim and Kondratiev, N and Kuznetsov, K and Trubin, V and Dimitrov, Lubomir,2018, Application of ultrasonic sensor for measuring distances in robotics, 1015,Journal of Physics: Conference Series, https://www.sciencedirect.com/science/article/pii/S0167739X13000241.

[3] Dunkels, Adam and Schmidt, Oliver and Voigt, Thiemo and Ali, Muneeb, Protothreads: Simplifying Event-Driven Programming of Memory-Constrained Embedded Systems,2006

[4] [Meyer&Mingins1999] Bertrand Meyer and Christine Mingins. Component-based Development: From Buzz to Spark. IEEE Computer, 32(7):35–37, July 1999.

[5] SOLIDWORKS TUTORIAL.1 SOLIDWORKS.(n.d). https://www.solidworks.de/sw/resources/10889_DEU_HTML.htm.

[6] [Galin2004] D. Galin, Software Quality Assurance: From theory to implementation. Harlow, England: Pearson Addison Wesley, 2004.

[7] [RUP1999] Ivar Jacobson, Grady Booch, and James Rumbaugh. The Unified Software Development Process. The Addison- Wesley Object Technology Series. Addison-Wesley, January 1999.

[8] Andrew P. Sage and James E. Armstrong. Introduction to Systems Engineering. Wiley Series in Systems Engineering and Management. Wiley-Interscience, March 2000.

[9] Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems By Bruce Powel Douglass

IX. Affidavit

Saikot Das Joy:

Eidesstattliche Erklärung

Hiermit bestätige ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen sowie Hilfsmittel genutzt habe. Alle Ausführungen, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind deutlich kenntlich gemacht. Außerdem versichere ich, dass die vorliegende Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Affidavit

I hereby confirm that I have written this paper independently and have not used any sources or aids other than those indicated. All statements taken from other sources in wording or sense are clearly marked. Furthermore, I assure that this paper has not been part of a course or examination in the same or a similar version.

Joy, Saikot Das	Hamm,16.07.2021	Saikot Das Joy
Name, Vorname	Ort, Datum	Unterschrift
Last Name, First Name	Location, Date	Signature

Manoj Luitel:

Eidesstattliche Erklärung

Hiermit bestätige ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen sowie Hilfsmittel genutzt habe. Alle Ausführungen, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind deutlich kenntlich gemacht. Außerdem versichere ich, dass die vorliegende Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Affidavit

I hereby confirm that I have written this paper independently and have not used any sources or aids other than those indicated. All statements taken from other sources in wording or sense are clearly marked. Furthermore, I assure that this paper has not been part of a course or examination in the same or a similar version.

Luitel, Manoj	Hamm,16.07.2021	
Name, Vorname	Ort, Datum	Unterschrift
Last Name, First Name	Location, Date	Signature

Stephen Eteng Patrick:

Eidesstattliche Erklärung

Hiermit bestätige ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen sowie Hilfsmittel genutzt habe. Alle Ausführungen, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind deutlich kenntlich gemacht. Außerdem versichere ich, dass die vorliegende Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Affidavit

I hereby confirm that I have written this paper independently and have not used any sources or aids other than those indicated. All statements taken from other sources in wording or sense are clearly marked. Furthermore, I assure that this paper has not been part of a course or examination in the same or a similar version.

Patrick, Stephen Eteng	Hamm,16.07. 2021	
Name, Vorname	Ort, Datum	Unterschrift
Last Name, First Name	Location, Date	Signature

Evrard Leuteu Feukeu:

Eidesstattliche Erklärung

Hiermit bestätige ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen sowie Hilfsmittel genutzt habe. Alle Ausführungen, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind deutlich kenntlich gemacht. Außerdem versichere ich, dass die vorliegende Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Affidavit

I hereby confirm that I have written this paper independently and have not used any sources or aids other than those indicated. All statements taken from other sources in wording or sense are clearly marked. Furthermore, I assure that this paper has not been part of a course or examination in the same or a similar version.

Evrard Leuteu Feukeu	Hamm,16.07.2021	
Name, Vorname	Ort, Datum	Unterschrift
Last Name, First Name	Location, Date	Signature

Aditya Kumar:

Eidesstattliche Erklärung

Hiermit bestätige ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen sowie Hilfsmittel genutzt habe. Alle Ausführungen, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind deutlich kenntlich gemacht. Außerdem versichere ich, dass die vorliegende Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Affidavit

I hereby confirm that I have written this paper independently and have not used any sources or aids other than those indicated. All statements taken from other sources in wording or sense are clearly marked. Furthermore, I assure that this paper has not been part of a course or examination in the same or a similar version.

Aditya Kumar	Lippstadt, 16.07.2021	
Name, Vorname	Ort, Datum	Unterschrift
Last Name, First Name	Location, Date	Signature

Muhammad Moaz Amin:

Eidesstattliche Erklärung

Hiermit bestätige ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen sowie Hilfsmittel genutzt habe. Alle Ausführungen, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind deutlich kenntlich gemacht. Außerdem versichere ich, dass die vorliegende Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Affidavit

I hereby confirm that I have written this paper independently and have not used any sources or aids other than those indicated. All statements taken from other sources in wording or sense are clearly marked. Furthermore, I assure that this paper has not been part of a course or examination in the same or a similar version.

Muhammad Moaz Amin	Hamm,16.07.2021	
Name, Vorname	Ort, Datum	Unterschrift
Last Name, First Name	Location, Date	Signature