IEEEDuino Student Contest 2024
Design Stage

Team Name:

IEEE Section: EGYPT

Team Members:

- Abdallah Ashraf
- Nour Mamdouh
- Mohamed Ashraf

University or Student Branch Name: CAIRO UNIVERSITY STUDENT BRANCH

Project Title: FUZZCAR

Project Scope, Purpose and Background

Due to the recent advancements in the medical field, doctors are facing multiple challenges related to handling samples to be analysed especially when speed and hygiene are premiums. This ensures that the doctor can have a well-organized database of all the samples he/she has. In many cases, this is not easy as there may be multiple samples for the same patient that need to be organized while maintaining a suitable level of sanitation for the surrounding environment as well as hygiene for the humans dealing with them especially if the samples are of hazardous nature.

Our project scope focuses on how to help doctors handle the medical samples and organize them without much human intervention, that is by applying control algorithms and database management strategies, we can effectively allow for little to no interaction between the doctor and the samples. We plan to construct a self-driven car that transfers samples from the doctor's position to the local storage repository. This is important because sometimes the doctor needs to closely monitor the patient after taking the required sample (blood) from him/her but needs to transfer the blood bag to the refrigerator. Thus, a conflict arises and hopefully we can solve this by using the self-driving car prototype which is far from being complete or ready to be implemented in real life but can provide a stepstone for the engineers working to develop such devices.

In addition to the self-driving semi-automated car, we plan to construct a 3D-printed robotic arm mounted on the car chassis that handles the sample during its transfer from the doctor's position to the repository (e.g. refrigerator). The robotic arm design is intended to be as straightforward as possible to minimize the need for human intervention such that the robot can grab the sample from the doctor easily without any disturbance to the normal workflow.

Design Criteria

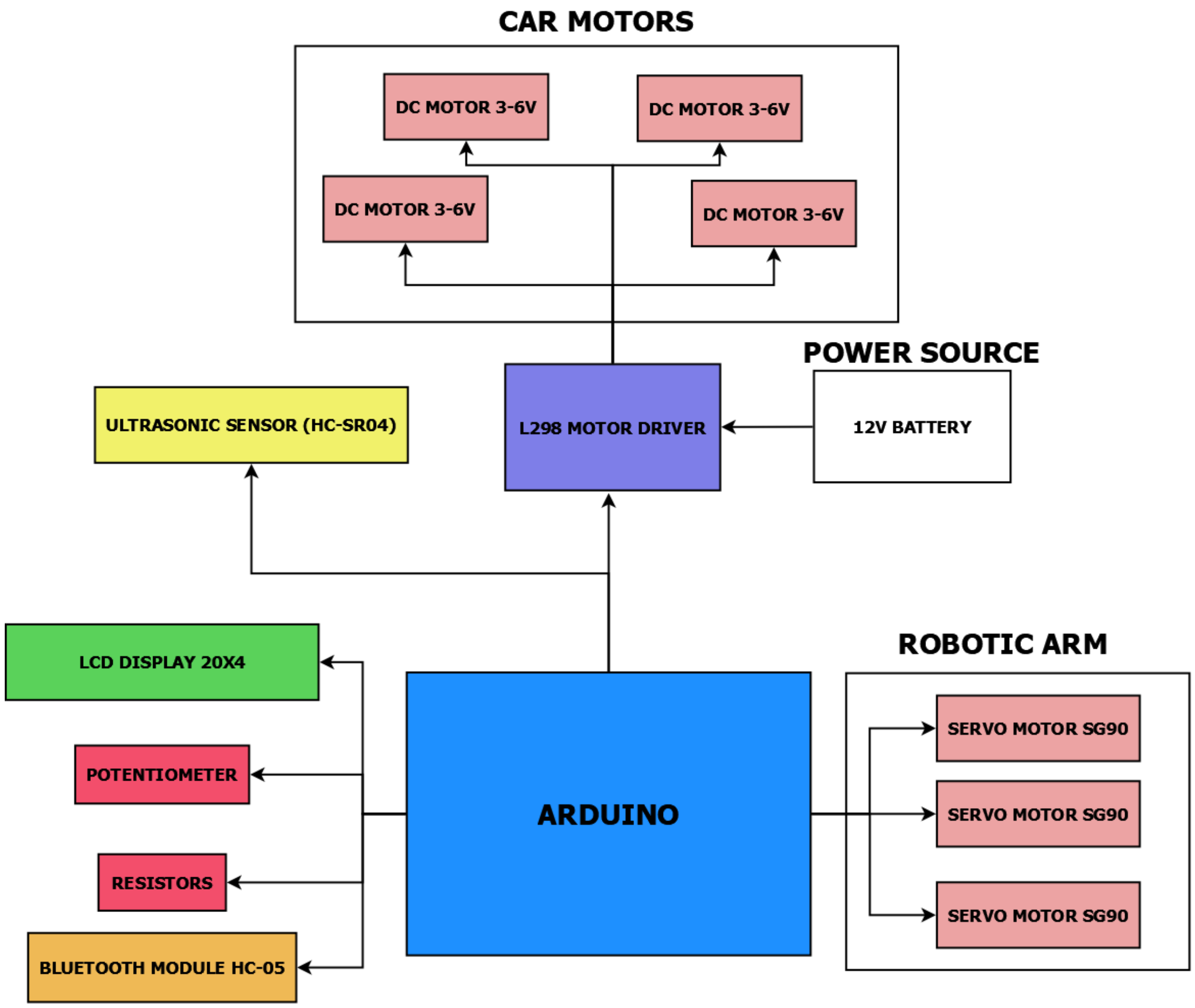
As we have already mentioned in the previous section, there are two main objectives for our project. The first being the self-driven car which moves on the set path which is not intended to be a complex one, in fact. Again, this is a mere prototype but gives the user a decent amount of control (e.g. the ability to stop or turn on the vehicle) over it. The second objective is the installation of a robotic arm structure that handles the object (in this case the medical samples). We are yet to determine the control strategy for the robotic arm meaning how it will grab/let go of the object in hand: whether it will be through proximity sensors or through direct order from the user.

The arduino nano microcontroller is a small but energy efficient device which allows for easy coding and has straightforward documentation. Since the scope of the project is inherently a type of machine learning controlled by previous knowledge of the surrounding envirmonment, there is much room for future development through the use of image processing and multiple device connectivity. However, due to the hardware limitations as well as the project scope, we are focused to deliver the main objectives stated above without the need for dealing with the complexities of intermediate robotics software.

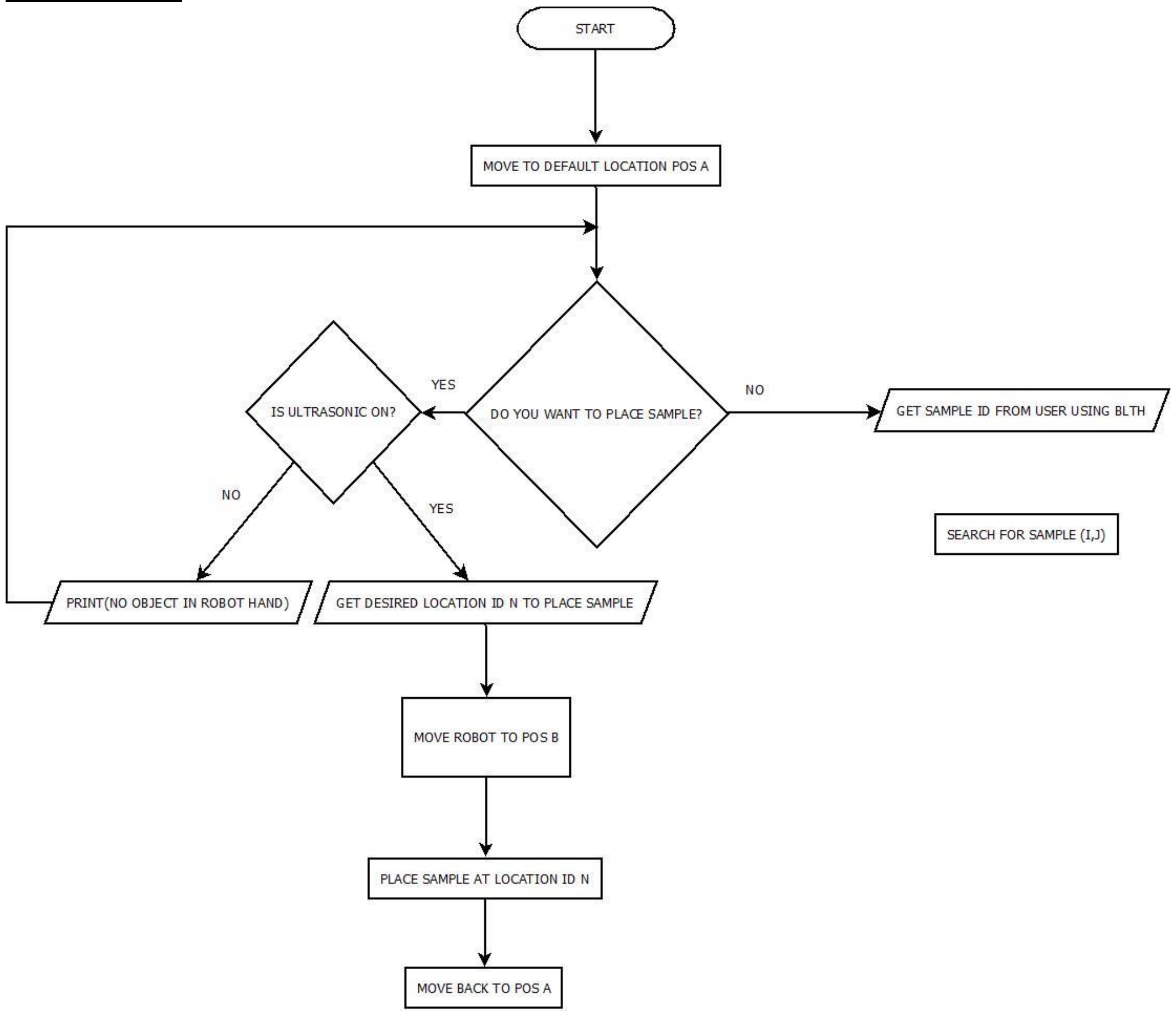
As for the sensors and actuators we plan to use, there will be main components namely: the dc motors and the drivers as well as the power supply (Li-ion batteries) for the car that will need to be properly sized as per the mechanical factors of the chassis and the object it will transfer, along with that there will proximity sensors for both the car to avoid obstacles as well as for the robotic arm to sense whether it holds the object or not. Optional features may be connecting the robot and the with a mobile application most probably could be a bluetooth controlled application.

Detailed Design

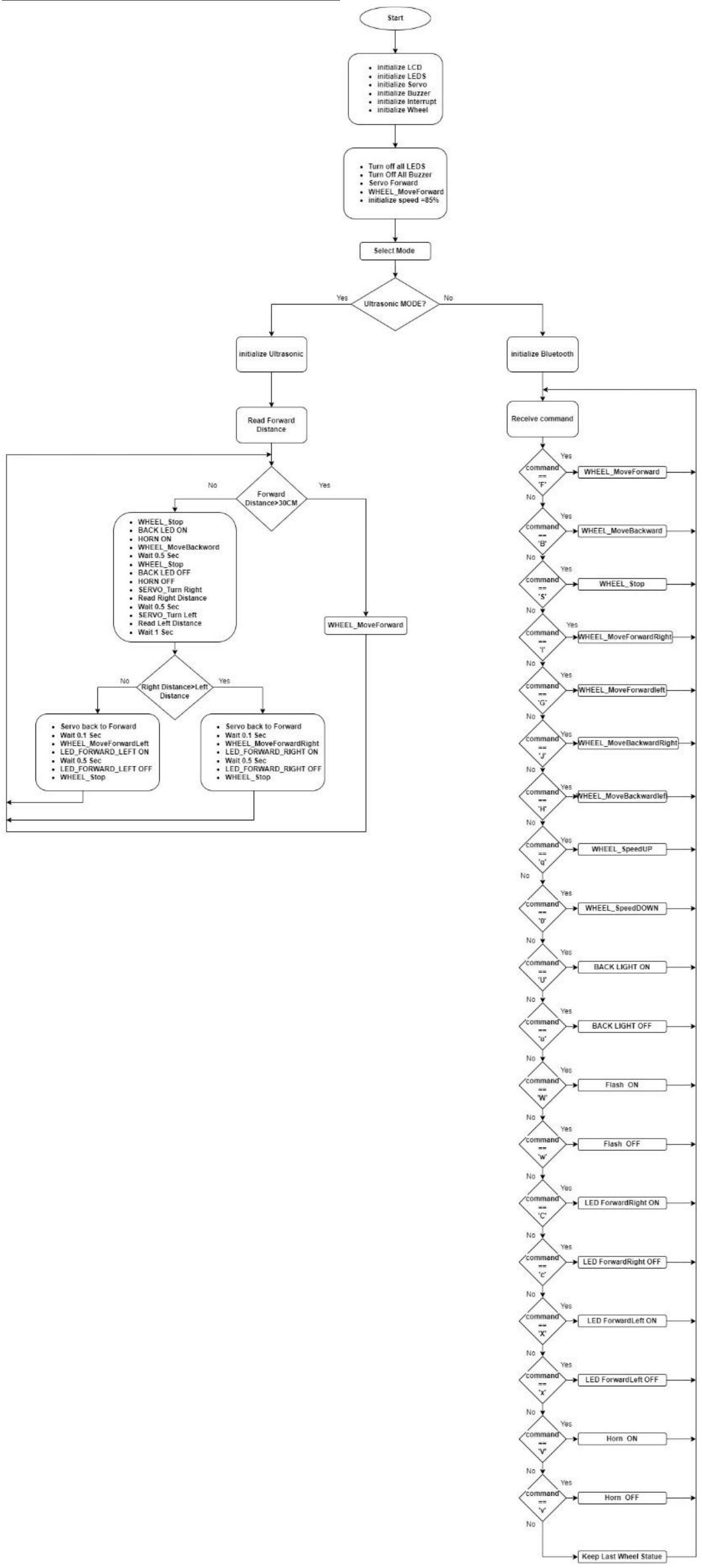
BLOCK DIAGRAM



FLOWCHART


START
MOVE TO DEFAULT LOCATION POS A
DO YOU WANT TO PLACE SAMPLE?
YES
NO
GET SAMPLE ID FROM USER USING BLTH
SEARCH FOR SAMPLE (I,J)
IS ULTRASONIC ON?
NO
YES
PRINT(NO OBJECT IN ROBOT HAND)
GET DESIRED LOCATION ID N TO PLACE SAMPLE
MOVE ROBOT TO POS B
PLACE SAMPLE AT LOCATION ID N
MOVE BACK TO POS A

EXTENDED FLOWCHART



<u>PSEUDOCODE</u>

<u>SIMPLE VERSION</u>

```
INT INITIAL_CARPOSITION = 0
INT X = GET (DESIRED CARPOSITION)
IF ROBOT ARM == FULL
        THEN Y = GET (DESIRED SAMPLE POSITION)
        MOVE ROBOT TO X
        PLACE SAMPLE IN Y
ELSE
        DISPLAY MESSAGE (NO OBJECT PLACED YET)
```

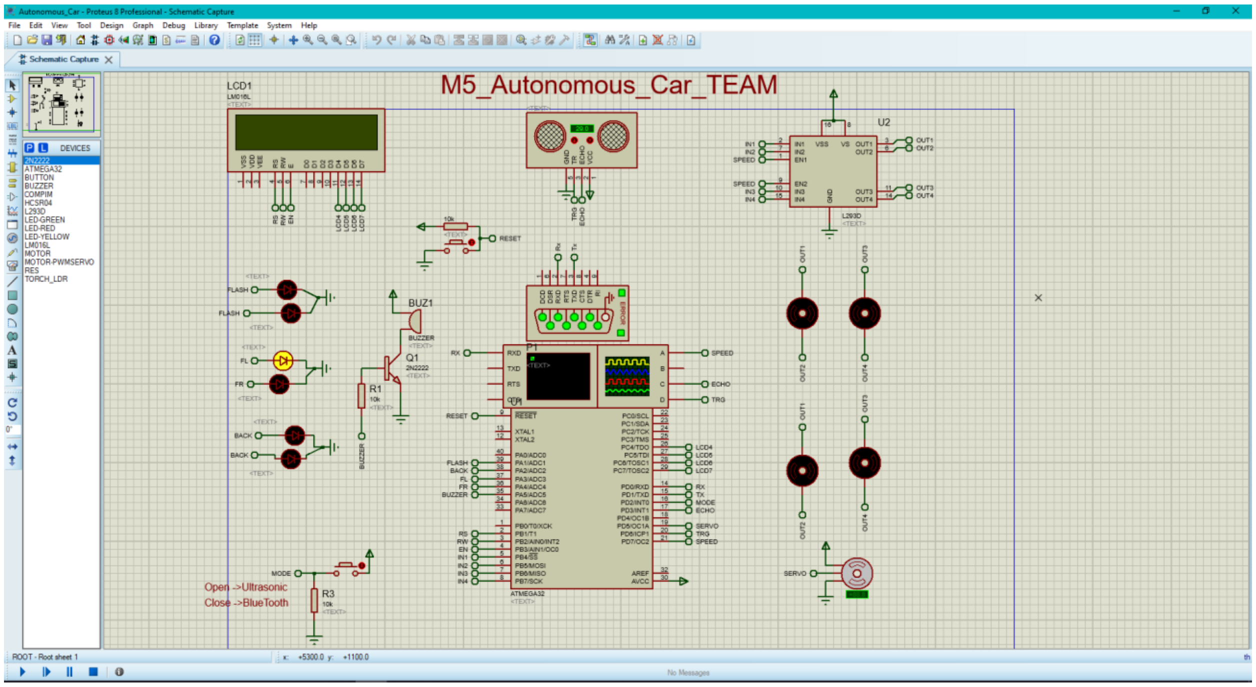
<u>EXTENDED VERSION</u>

```
 DECLARE position_Vangle[6] SET TO [90 180 270]
   DECLARE position_Hangle[6] SET TO [90 180 270]
   DECLARE Position_ID[6] SET TO [1 2 3 4 5 6]
   DECLARE ISITFREE[6] SET TO [-1 -1 -1 -1 -1 -1]

   DECLARE direction (integer: 1 for forward, -1 for backward)
   DECLARE targetSpeed SET TO 120

   DECLARE ID (integer)
   if start is PRESSED THEN  // if start button is PRESSED
      {
   setDCmotor(motorPinA,motorPinB,-1,120)   // move the stick to the original place
   setServoAngle(servoPin1,0)
   setServoAngle(servoPin2,0)

   //if Dr. prssed place sample and Ultrasonic detected a sample present
    if (PLACE_SAMPLE is PRESSED) AND (ULTRA_SONIC is True) THEN
       {
       count = 0  // to check the sample found empty place

          FOR i = 1 TO 6   // to loop on the positions till an empty one is found
          if (ISITFREE[i] = -1 ) THEN
             // move motors to the position of the empty slot
             setDCmotor(motorPinA,motorPinB,1,120)
             setServoAngle(servoPin1,position_Vangle[i] )
             setServoAngle(servoPin2,position_Hangle[i] )
             // state it is not free anymore
             ISITFREE[i]=1
             // quit looping
             Exit FOR
           Else
```

```
              ENDIF
              count = count +1
             NEXT i
             // check if no empty slot is found
             IF count = 5 THEN
                 print ("no available places")
            ENDIF
          setDCmotor(motorPinA,motorPinB,-1,120)   // move the stick to the original place
          setServoAngle(servoPin1,0)
          setServoAngle(servoPin2,0)
         }

        if (GET_SAMPLE is PRESSED)  //IF dr requested to get a sample
            {
            count = 0 // to check if the requested sample is found
                ID <- INPUT("Enter Sample ID: ")  // enter the ID of the requested sample
                  FOR i = 1 TO 6
                      if (Position_ID[i] = ID) THEN // to know the position of this ID
                          {
                              if (ISITFREE[i] = 1) THEN // to check if this position has a sample
                                  {
                                   // move motors to the position of the sample slot
                                     setDCmotor(motorPinA,motorPinB,1,120)
                                     setServoAngle(servoPin1,position_Vangle[i] )
                                     setServoAngle(servoPin2,position_Hangle[i] )
                                    // wait 1 sec
                                     delay_in_ms(1000)
                                     // return to the original place
                                     setDCmotor(motorPinA,motorPinB,-1,120)   // move the stick to the
original place
                                     setServoAngle(servoPin1,0)
                                     setServoAngle(servoPin2,0)
                                     // update the array as this position is free now
                                     ISITFREE[i] = -1
                                     // quit looping
                                     Exit FOR
                                  }
                          }
                          count = count + 1
                   NEXT i
             IF count = 5 THEN // check if this sample ID is not found
                 print ("WRONG ID")
            ENDIF
            }
```

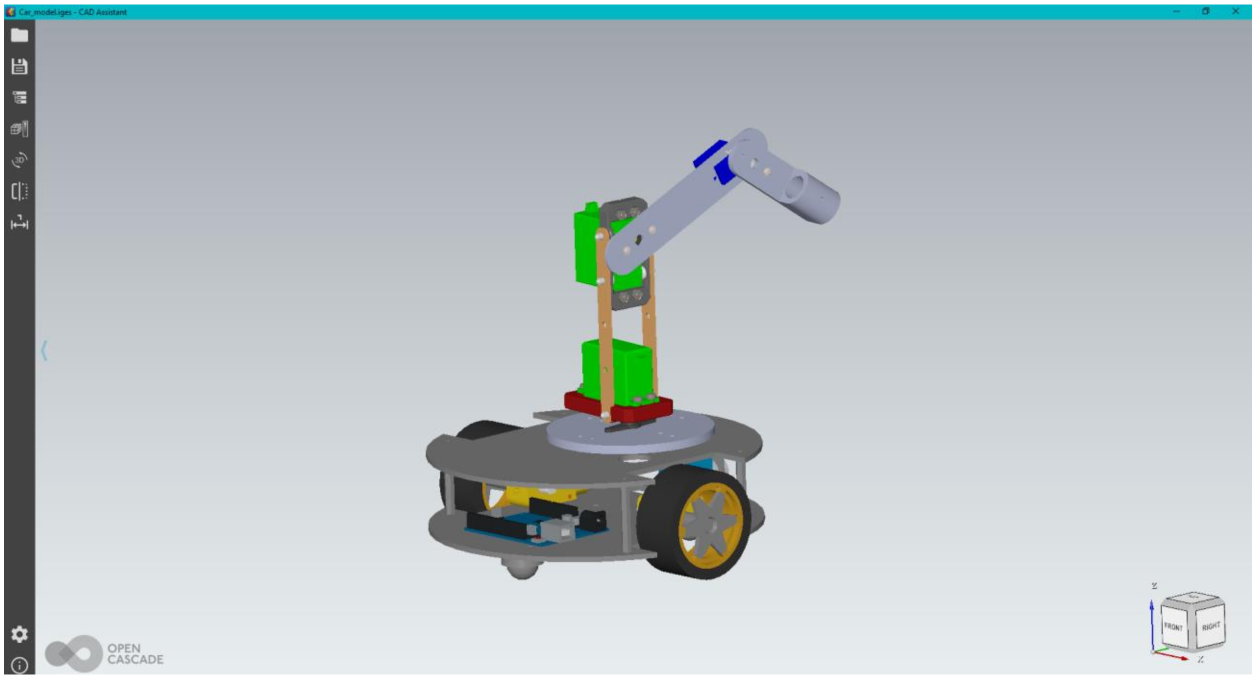
```
     }

FUNCTION setDCmotor(motorPinA, motorPinB, direction, targetSpeed)

  // Set motor direction
  IF direction == 1 THEN
    WRITE_HIGH(motorPinA)
    WRITE_LOW(motorPinB)
  ELSE
    WRITE_LOW(motorPinA)
    WRITE_HIGH(motorPinB)
  ENDIF

  // Apply PWM control for speed (similar to setDCmotorSpeed function)
  dutyCycle = map(targetSpeed, 0, maximum_speed, minimum_duty, maximum_duty)
  CONFIGURE_PWM(motorPinA, dutyCycle)  // Adjust pin depending on direction

ENDFUNCTION


FUNCTION setServoAngle(servoPin, targetAngle)

  // Convert angle to pulse width (adjust conversion based on servo specifications)
  pulseWidth = map(targetAngle, 0, maximum_angle, minimum_pulse, maximum_pulse)

  // Send pulse signal to servo pin
  WRITE_PULSE(servoPin, pulseWidth)

ENDFUNCTION
```

CIRCUIT LAYOUT (USING PROTEUS SOFTWARE)



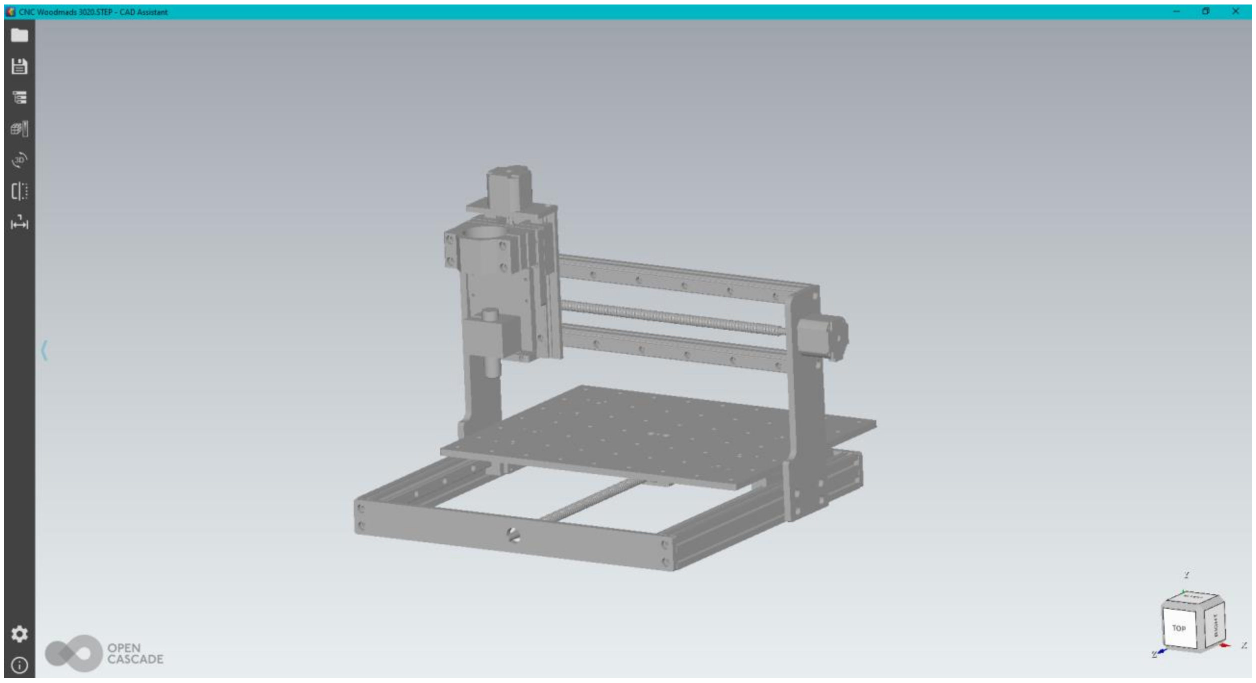
<u>SUGGESTED 3D MODELS FOR THE ROBOTIC ARM AND THE CAR</u>
These models are conceptual designs and will be further customized according to the topology of the final project and the features required.

CONCEPTUAL DESIGN 1



<u>A two wheeled car with a robotic arm having 3 degrees of freedom.</u>

CONCEPTUAL DESIGN 2



<u>A robotic arm with a design similar to CNC machines.</u>

Forward Plan

We will test the proposed design by...

1- ASSEMBLY
   First we need to assemble the car and the robotic arm mechanical components ensuring proper fixation, lightweight and efficient space usage.
   Secondly, we must wire the circuitry with the Arduino nano as well as the sensors.
   The sensors include but are not limited to ultrasonic and Bluetooth module.
   Thirdly, we need a power supply for our robot. In most similar cases, hobbyists use Li-ion batteries found in Laptop computers as they combine light weight and high energy density.
2- CODING:
   The term coding means setting the control algorithm for the dc motors to steer the car as well as the robotic arm structure.
3- TESTING ensures that we have achieved the intended role of the project namely to move along a set path whether with or without human intervention. This is divided into several stages:
   1- DC motors dunctioning properly
   2- The car steers both manually and automatically.
   3- The robotic arm can take and place objects from the starting position to the desired location.
   4- The robot can correctly sort the objects according to the callback which is from the user. (e.g. put object x in position y)