Aim: - The aim of this Project is to create an autonomous robot capable of playing a game of soccer using a spray painted squash ball

Objectives:-

1) Design a robot to be used specifically for the field provided

2) Develop a search ball function for the robot to locate balls on the field and capture them

3) Develop a wall detection and collision avoidance function so the robot does not bump into the walls

4) Develop a goal scoring function where the robot searches for the torch light to determine its target and then shoots the ball

Introduction:-

The aim of this Project was to create an almost autonomous soccer robot. In order to achieve this goal the robot must be capable of perception of the environment (field) and respond with appropriate action. The robot received assistance from a torch in order to aim it towards the goal which is why the term "almost autonomous" is used. There is some degree of uncertainty involved in the various actions taken by the robot for example the robot can miss the goal at times if there was a slight difference in angle at the point of contact between the robot and the ball while shooting. The main behaviors that the robot should be capable of are maneuvering around the field, searching for balls and scoring goals. These behaviors or actions of the robot transition between each other when certain conditions are met for example when the robot sees the ball it stops doing the "search for ball" function and starts performing "get ready to shoot" function.

The competitive environment created by this competition aims to bring together skills such as autonomous robot design, design for specific functions, artificial intelligence and programming. It encouraged innovation from the students in order to achieve the various objectives of the assignment.

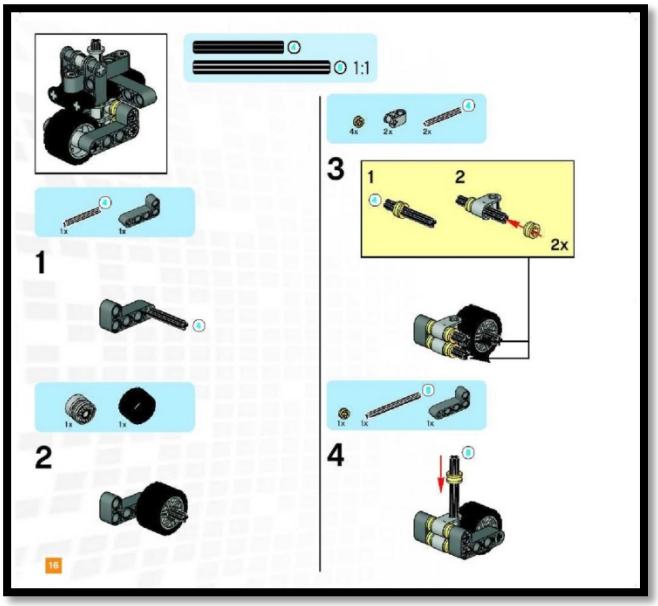
Design:-

Total Sensors used: - 3

Detecting ball: - Ultrasonic sensor and light sensor

Detection of wall: - Ultrasonic sensor

Detection of light beam from torch: - light sensor



We re-used the same caster wheel (Figure 1) from the previous assignment as it proved to be very effective while maneuvering the robot.

Figure 1

We had to completely change the design in order to accommodate the additional sensors and the motor on top of the robot that would be used to operate the cage and kicker. We decided to keep our design small and simple due to the additional motors and sensors and the power requirements. We figured that if the robot was too bulky it would be too slow due to its size and the high consumption of power from the motors and all the sensors. We decided to build a cage that could kick using the bottom part of its back wall to make the kicking process faster and easier to implement.

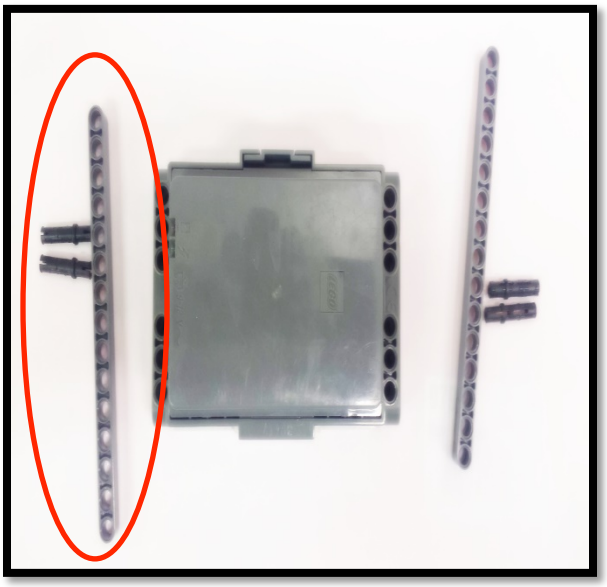


Image 1

Image 1

Attach 2 beams (size 15) to the main mindstorms nxt unit using 4 large black pegs/pins.

Rationale: These beams will be used for the attachments in the front of the robot like the light sensor

Image 2

Take 2 angle connectors, 1 bush, and 2 half bushes and an axle and make the attachment as illustrated in the picture

Rationale: The circled attachment will be used later on to attach the motors on both sides to the robot

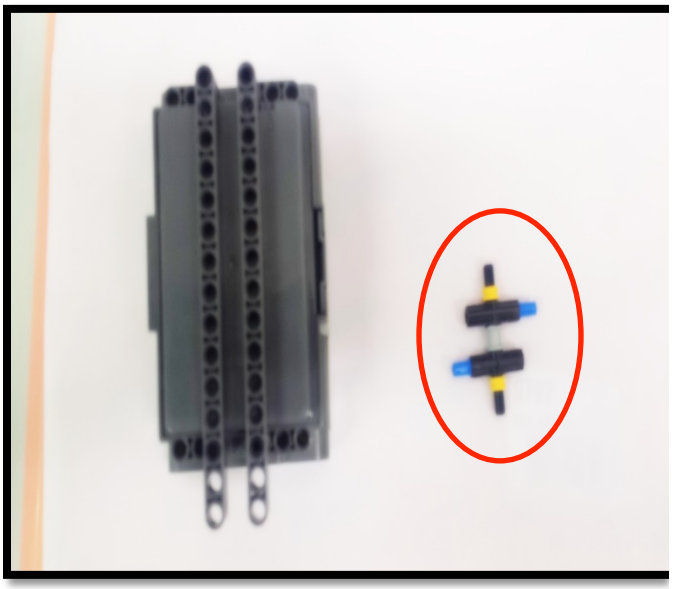


Image 2

Image 3

This picture illustrates the motor attachments required to fix the wheels onto the main unit

Prepare an axle and two bushes to use for the wheel as illustrated on the left part of the image.

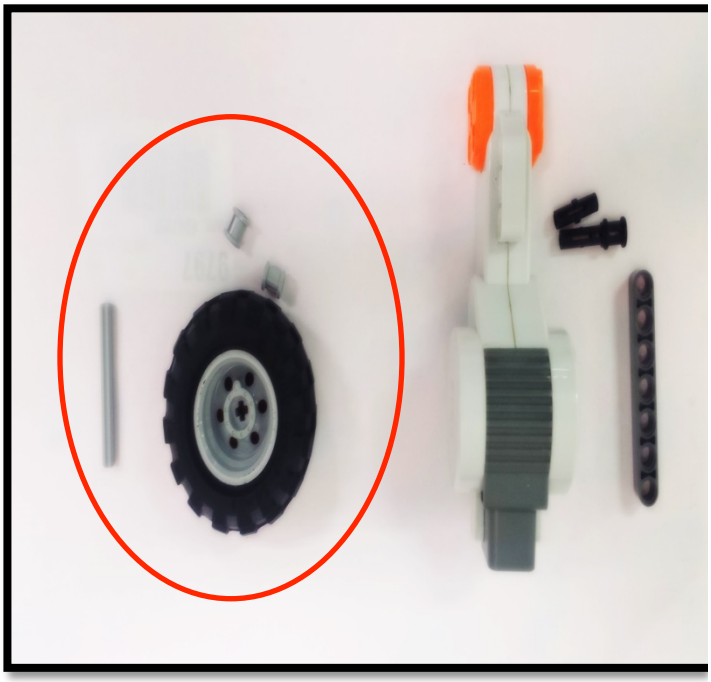


Image 3

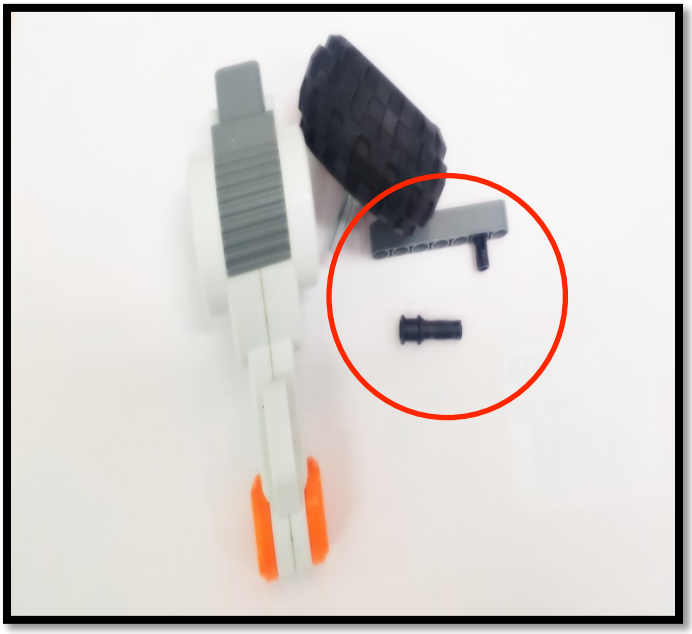


Image 4

Image 4

Prepare a size seven beam with two pegs/pins as illustrated in the image

Image 5

Attach the parts gathered in image 3 and image 4 as illustrated in image 5

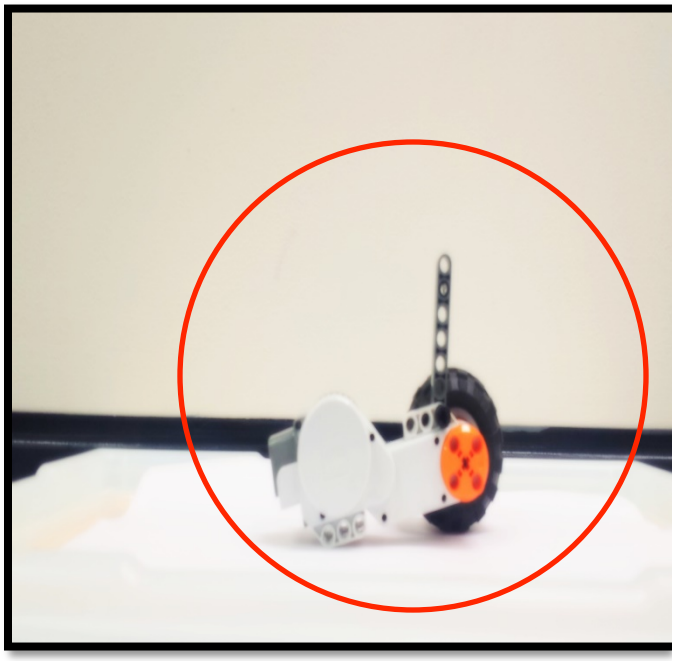


Image 5

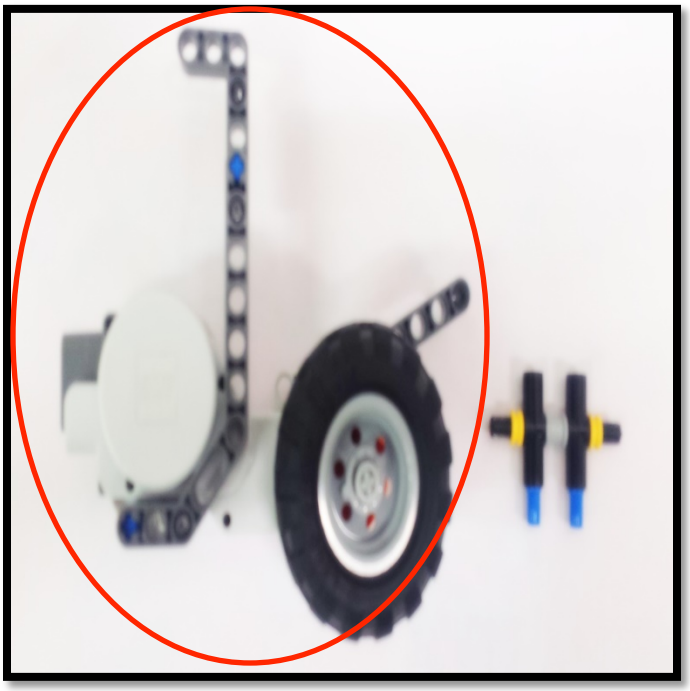


Image 6

Image 6

Next you need to create the base attachment of the motor as shown on the pic which is illustrated in the following images as well as the attachment prepared in image 2 which will be used to attach the two motors on the base unit

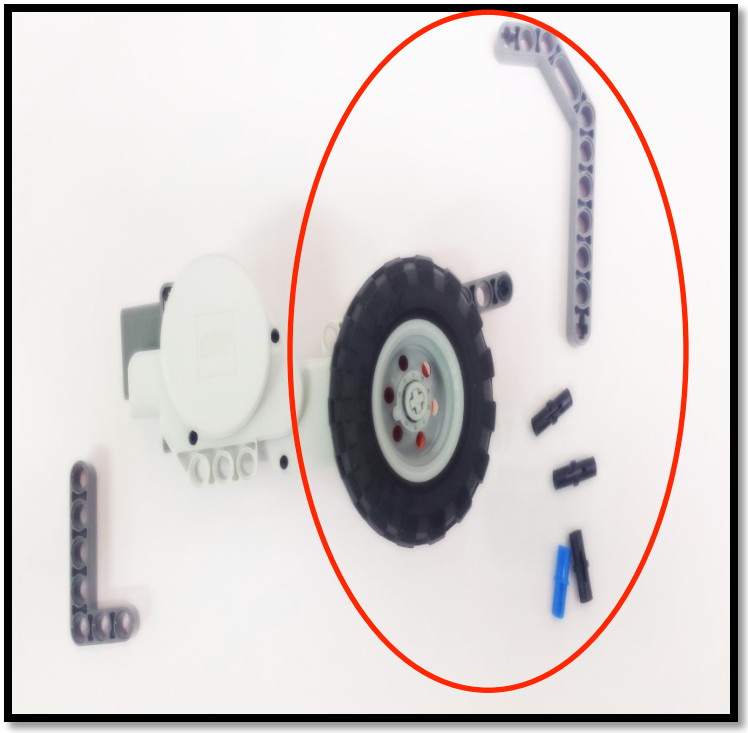


Image 7

Image 7

In order to create the attachment in image six gather Lift Arm bent 45 angled beam, 3 pegs/pins and an axle pin (blue/cream)

Image 8

Fix the attachment using the pieces gathered in image 7 to give the structure in image 8

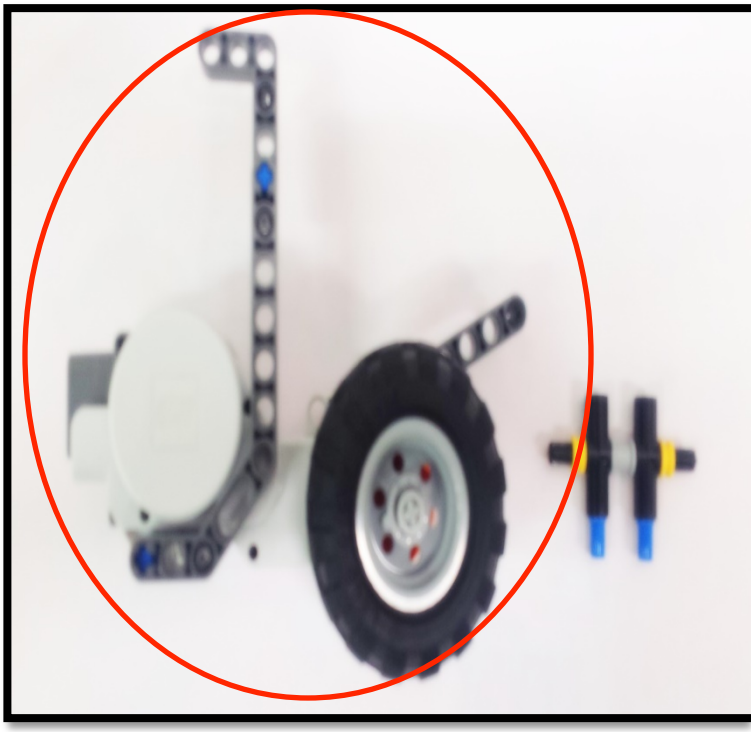


Image 8

.

Image 9

Image 9

Make a laterally inverted copy of the structure shown in image 8 for the other side

Image 10

Image 10

Attach the 3 sections shown in image 9 in order to get the structure illustrated in image 10

Image 11

This illustrates the ultrasonic sensor attachment. Gather 2 inverted cross black pieces, an axle of appropriate length and peg/pin attachments

Rationale: - We found it best to attach the ultrasonic sensor in the front close to the ground so that it can see the balls clearly during the scans

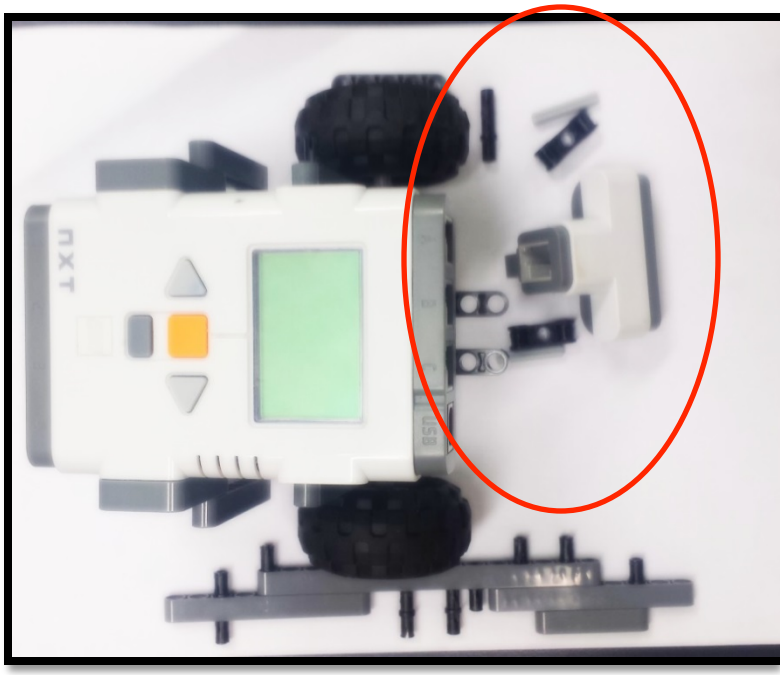


Image 11

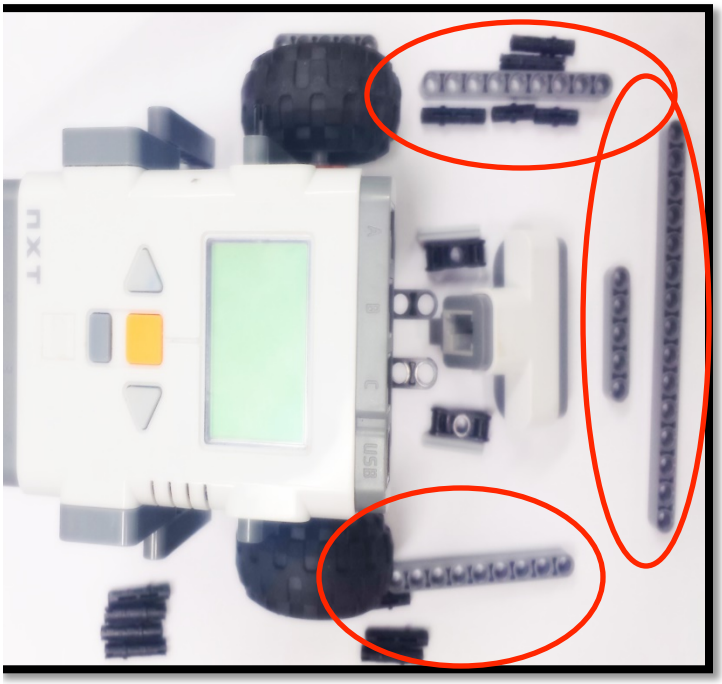


Image 12

Image 12

Gather beams and pegs to create something similar to a mud guard for the tires. The length of this can be varied to preference.

Rationale: - This stops the tires from hitting a ball and getting stuck

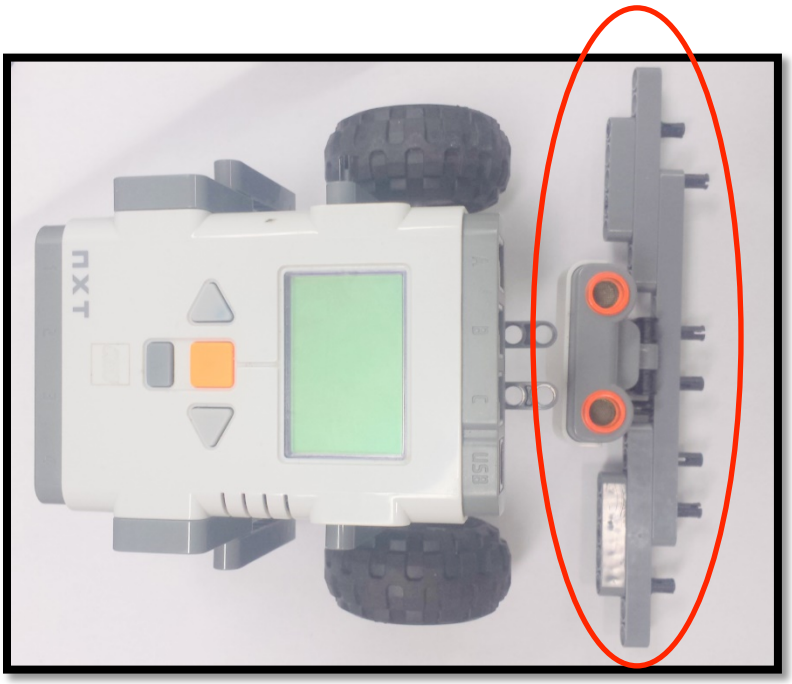


Image 13

Image 13

Illustrates the final structure created from the parts gathered in image 11 and image 12

Image 14

Image 14

Gather two perpendicular angled beams, 4 pegs/pins and two pegged blocks as illustrated in the image and fix it on top of the base unit as shown in image 15

Image 15

Gather a Lift Arm bent 45 angled beams, a light sensor and a peg. This will be used to attach the light sensor on the side of the robot

Rationale: The light sensor is fixed at the side of the robot with appropriate height in order to get a clear beam from the torchlight without any interference

Image 15

Image 16

Front view of the robot so far

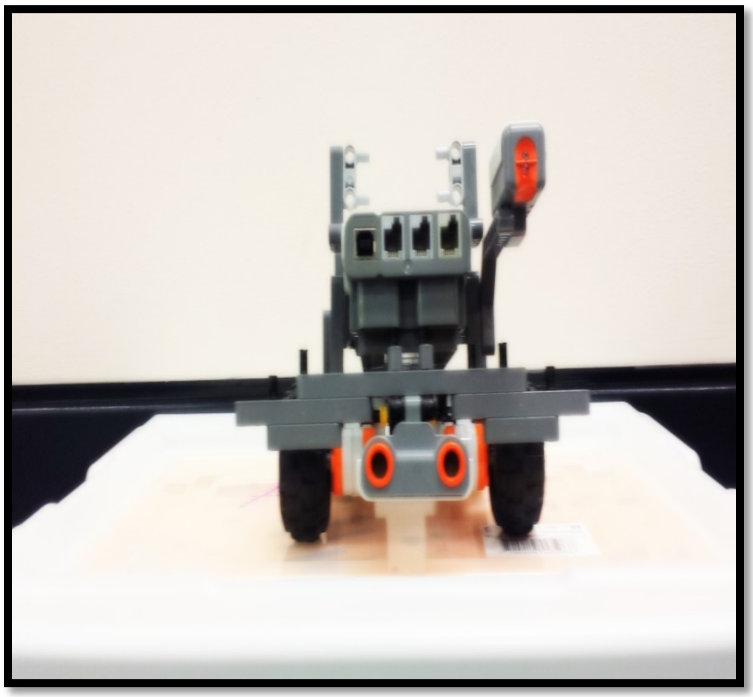


Image 16

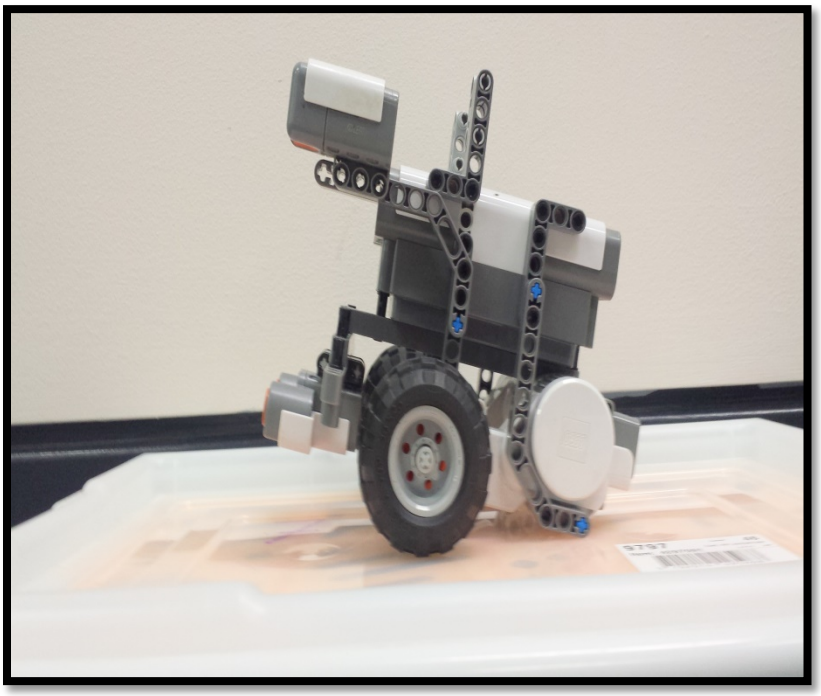


Image 17

Image 17

Side view of the robot so far

Image 18

Rationale: - Now we are going to illustrate the building of the cage. This is used both to catch the ball and to shoot the ball

Gather 2 Lift Arm bent 45 angled beams, 4 pegs and 2 axle pins

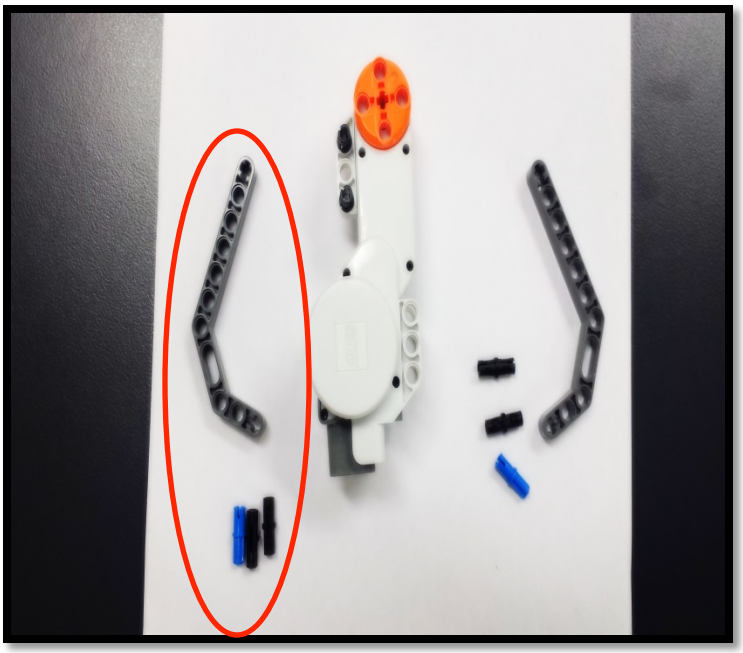


Image 18

Image 19

Fix the parts gathered in image 18 as shown in image 19

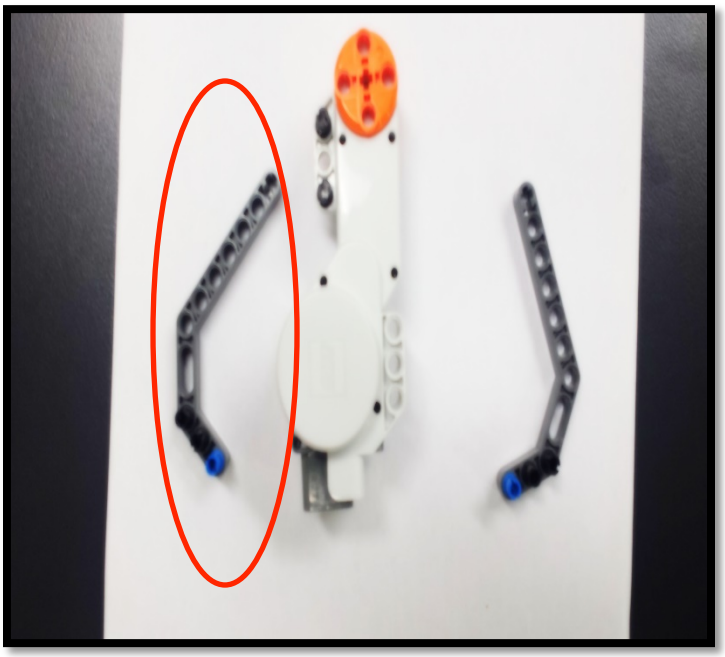


Image 19

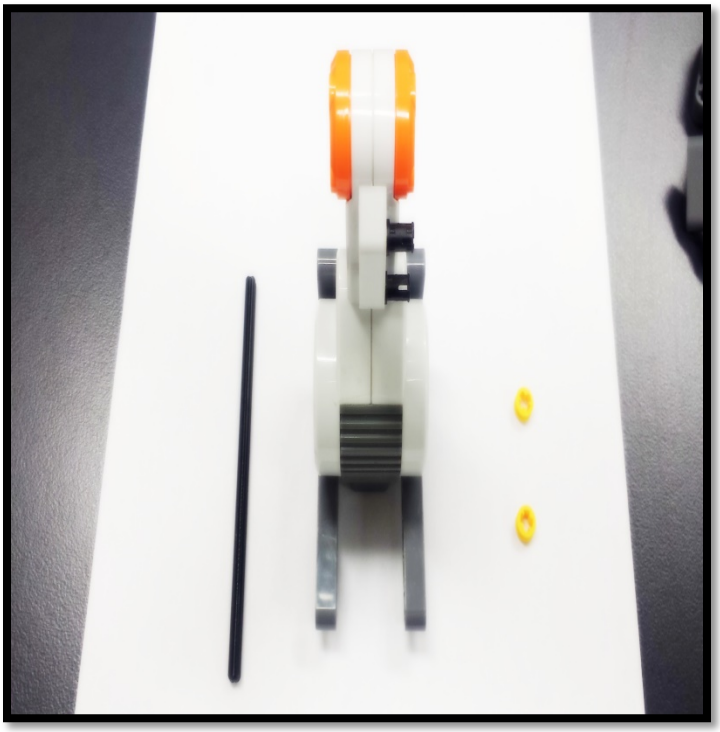


Image 20

Image 20

Gather an axle and two half bushes

And put it through the motor

Image 21

Put the axle through the motor as illustrated in image 21

Gather 4 angled beams, two small beams, axle pins and pegs as illustrated in this picture.

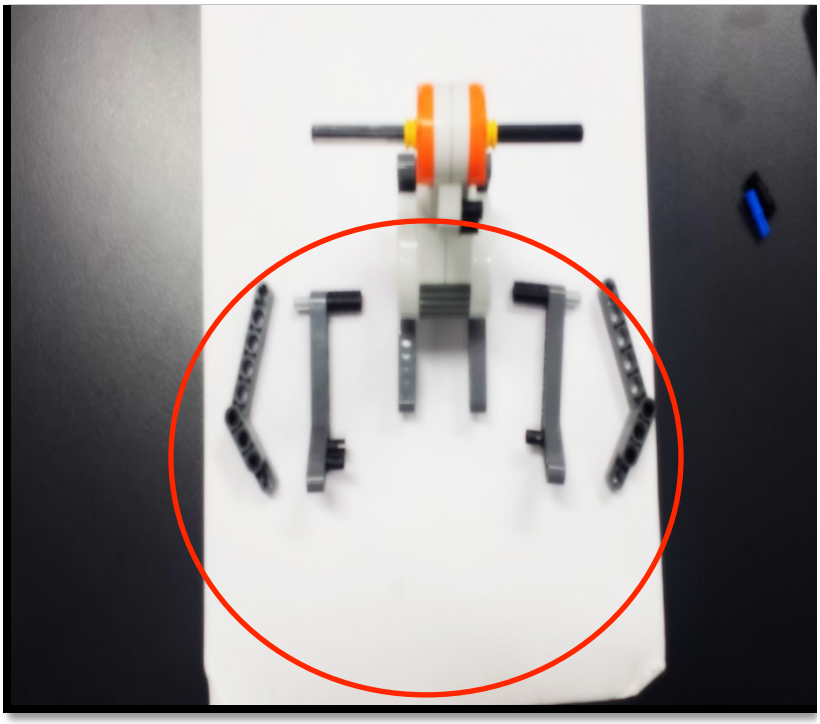


Image 21

Image 22

Use the parts gathered in image 21 and attach it as illustrated in image 22

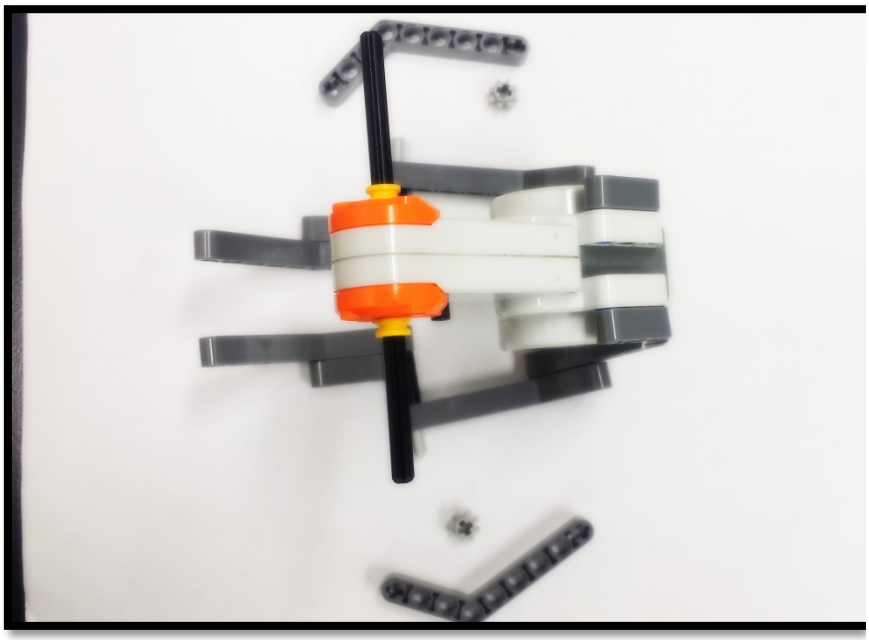


Image 22

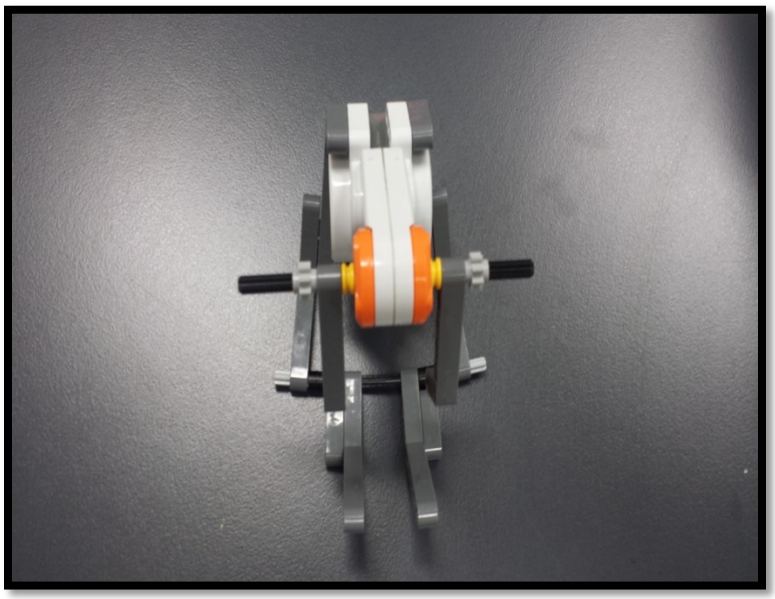


Image 23

Image 23

Image 23 illustrates the base of the cage that is going to be used to capture the ball and shoot it

The back wall of the cage had too many gaps

Which need to be filled up?

Image 24

We filled the gaps in image 23 using whatever beams were available to us. The parts were not ideal but since we ran out of the parts we needed we had to use whatever was available to us.

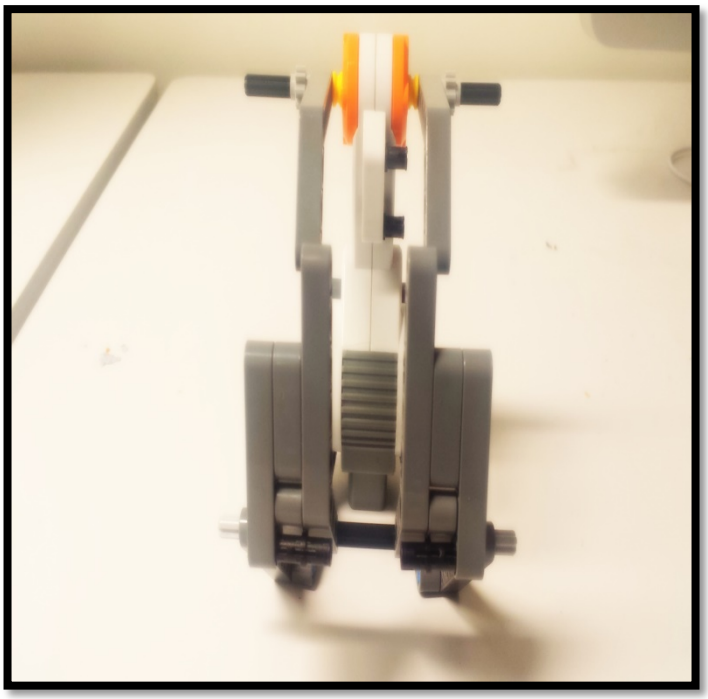


Image 24

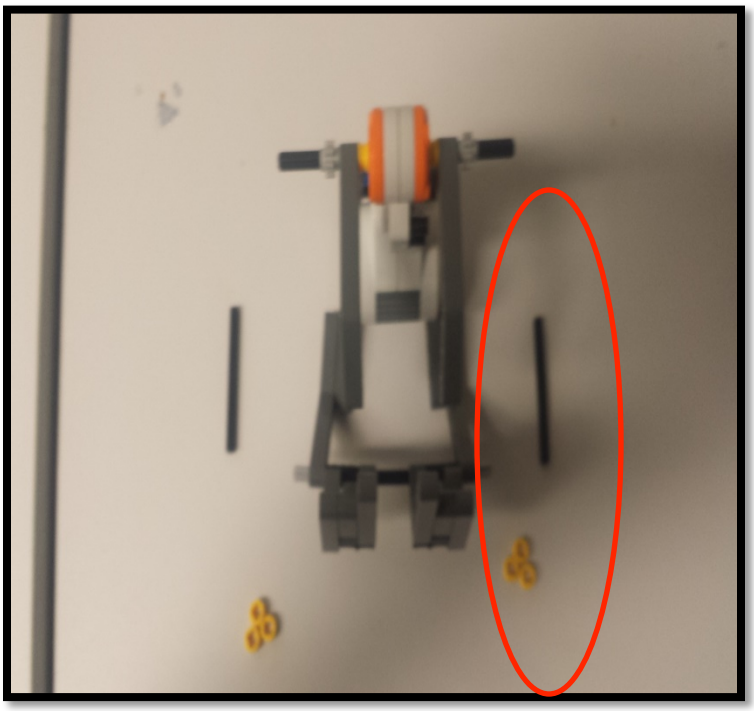


Image 25

Gathers two axles and bushes or half bushes to tighten the axle

Image 25

Image 26

We used multiple half bushes here because we ran out of full bushes

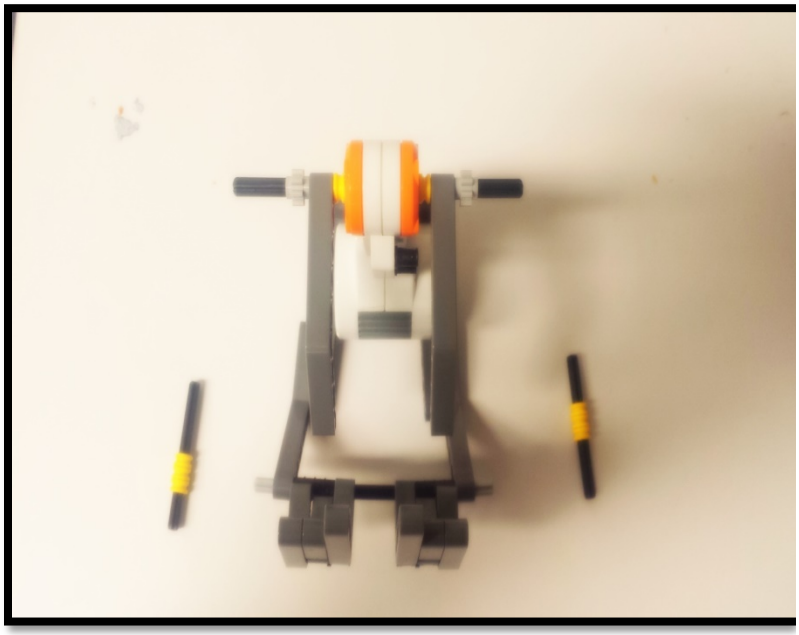


Image 26

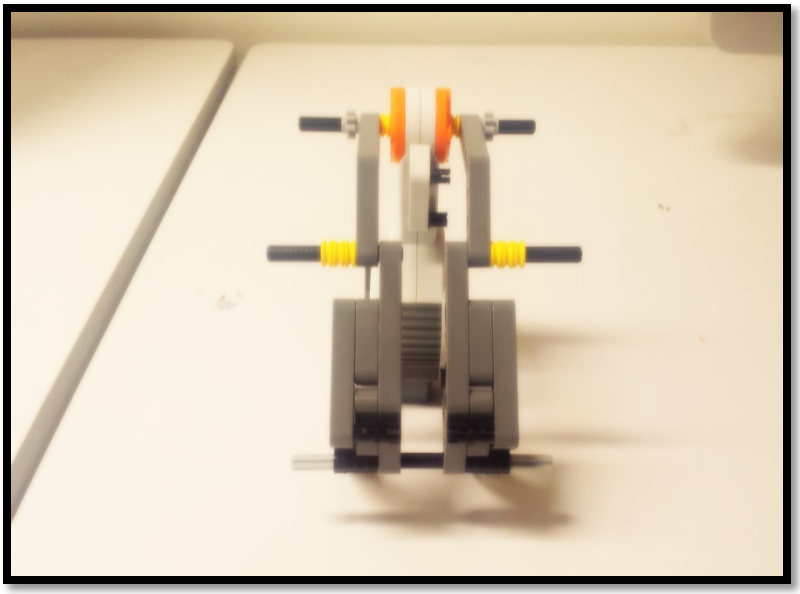


Image 27

Image 27

The structure we get of the cage so far after the attachments from image 25

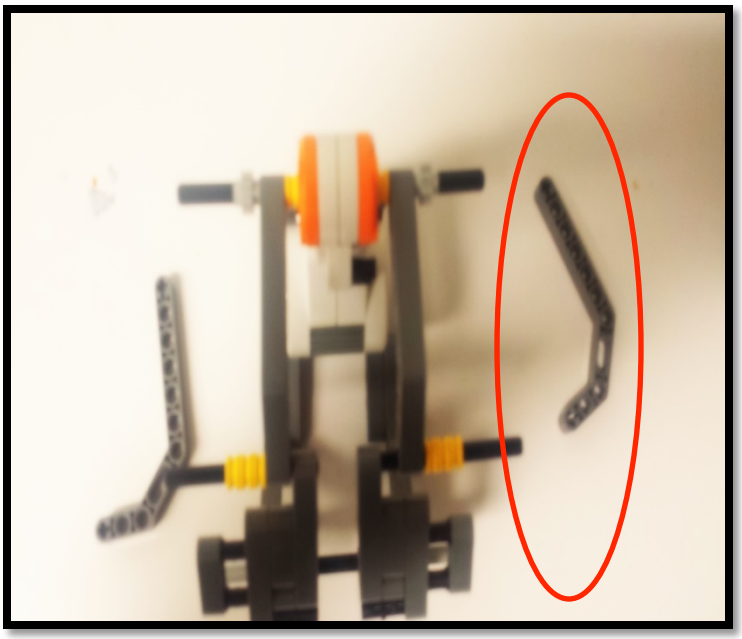


Image 28

Image 28

Next we have to start building the arms and enclosure of the cage.

Gather 2 Lift Arm bent 45 angled beams

Image 29

Attach the angled beams using appropriate pegs and axle pins as illustrated in Image 29

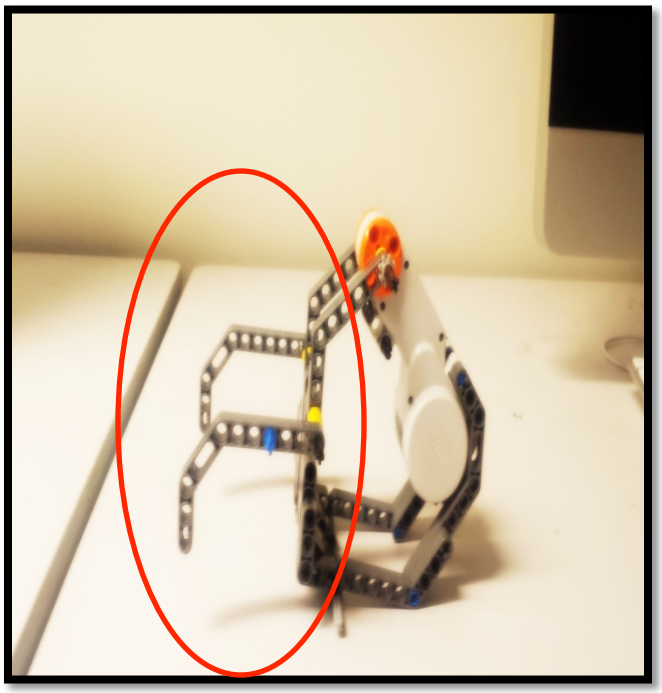


Image 29

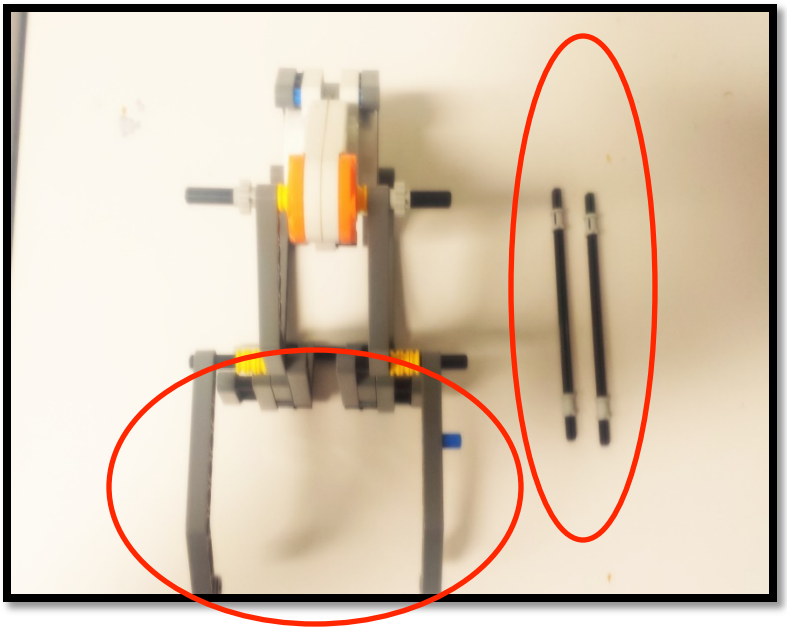


Image 30

Image 30

Using axles and bushes we need to close the cage in the front in order to capture the ball

Image 31

This is the structure we get after attaching the parts shown in image 30

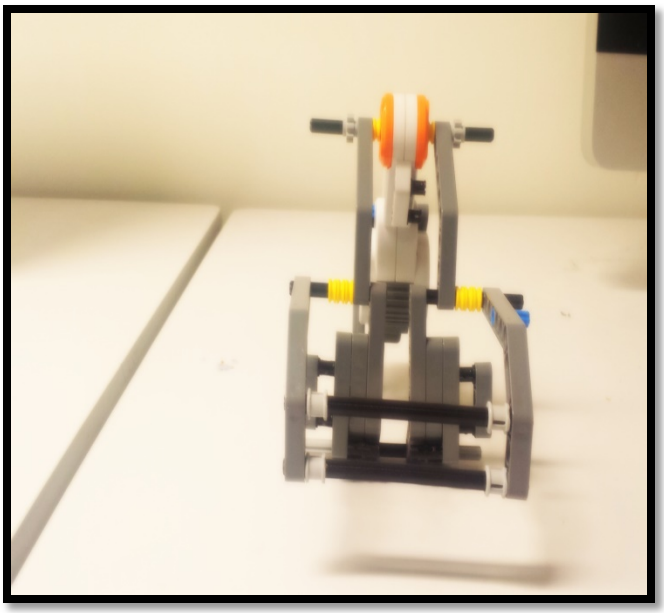


Image 31

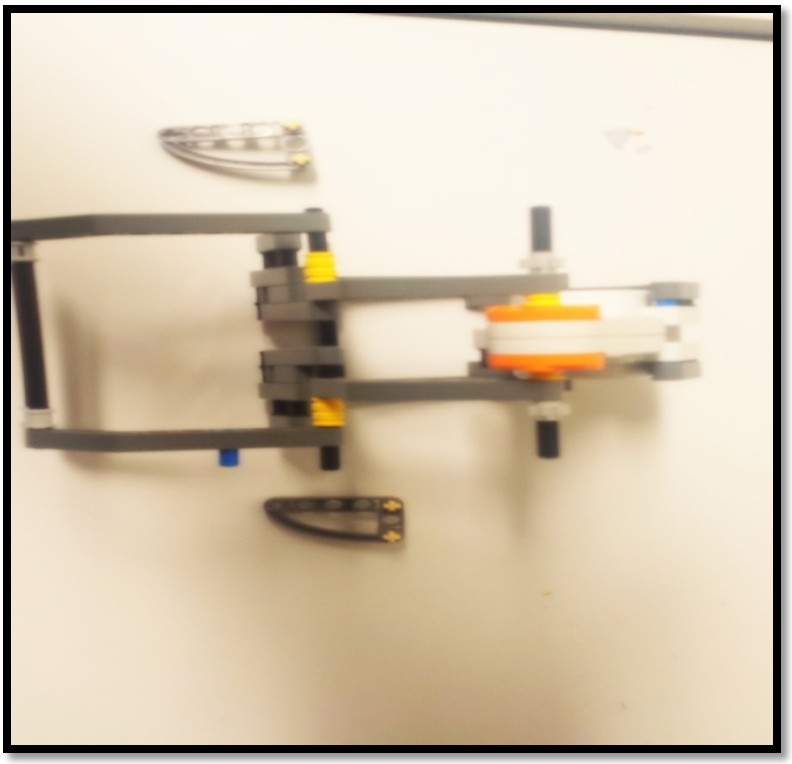


Image 32

Image 32

Next we need to complete the enclosure on the sides to finish the cage.

We use two Liftarm Bent 90 Quarter ellipse’s and axle pins for this.

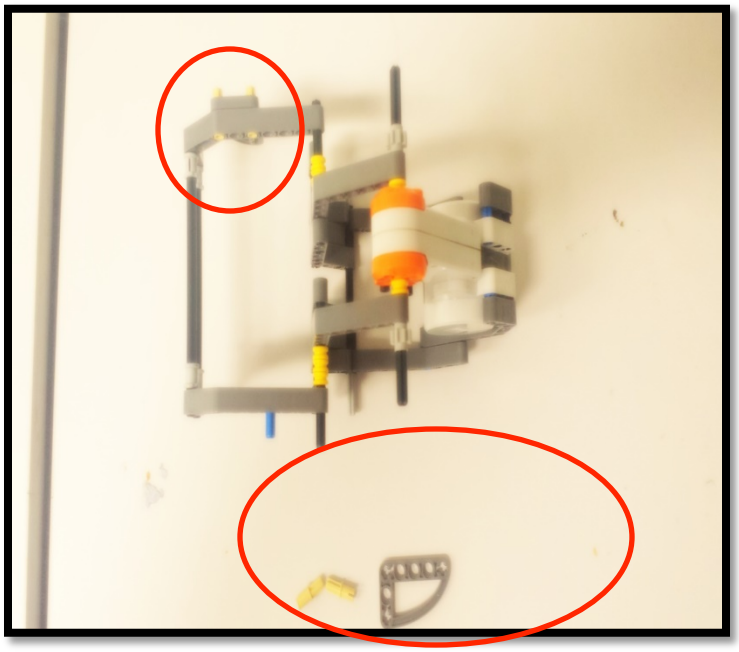


Image 33

Image 33

Attached the parts gathered in image 32 as illustrated in image 32

Image 34

This image illustrates the finished cage after all the attachments

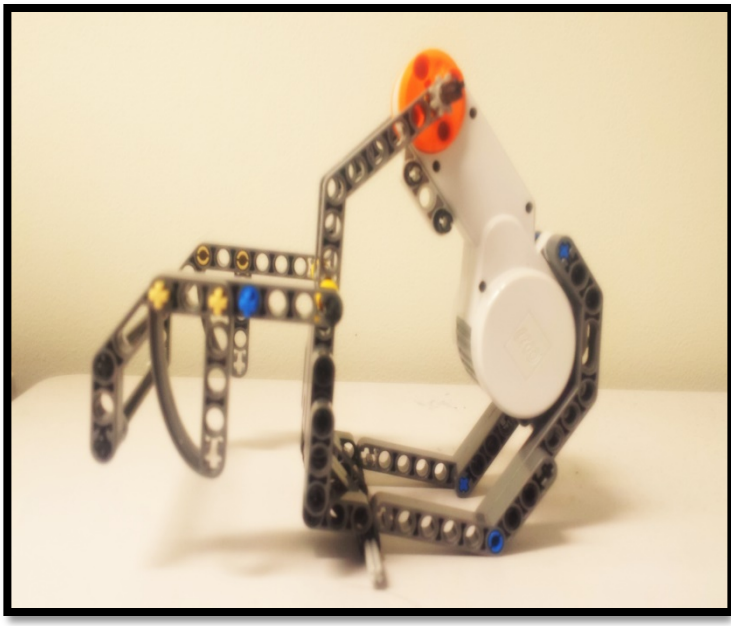


Image 34

Image 35

Finally we needed to fix an additional light sensor for the ball detection algorithm. We ran out of angled beams and the required parts to get it in the position we wanted so we had to tape it on the ultrasonic sensor

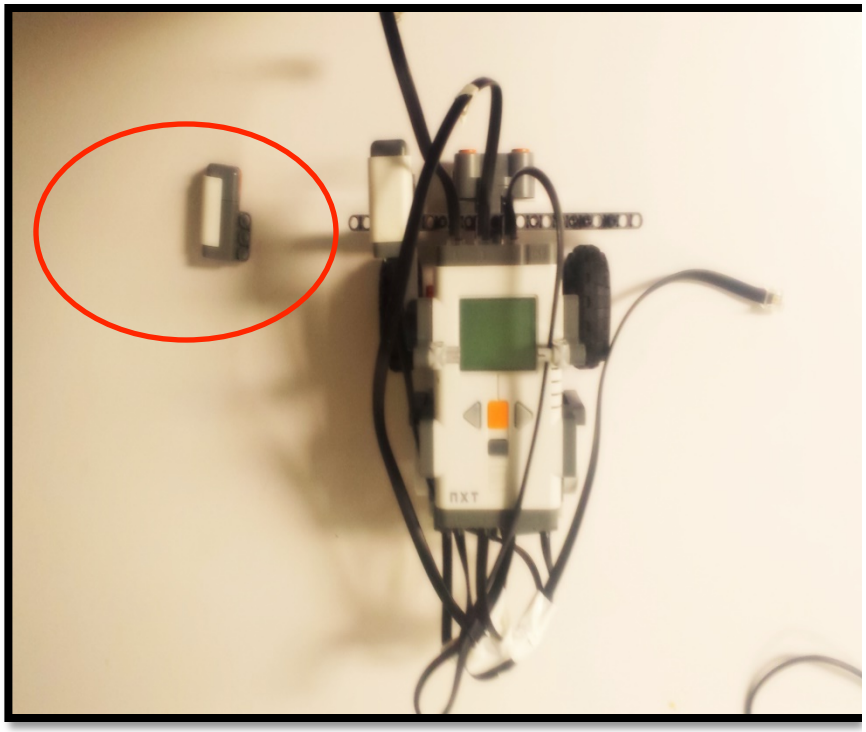


Image 35

Image 36

We used sticky tape on the cage in order to reduce the friction as the pain of certain balls had worn off and was getting stuck in the cage preventing the ball from moving forward.

After the cage is attached to the robot it givens the final structure below.

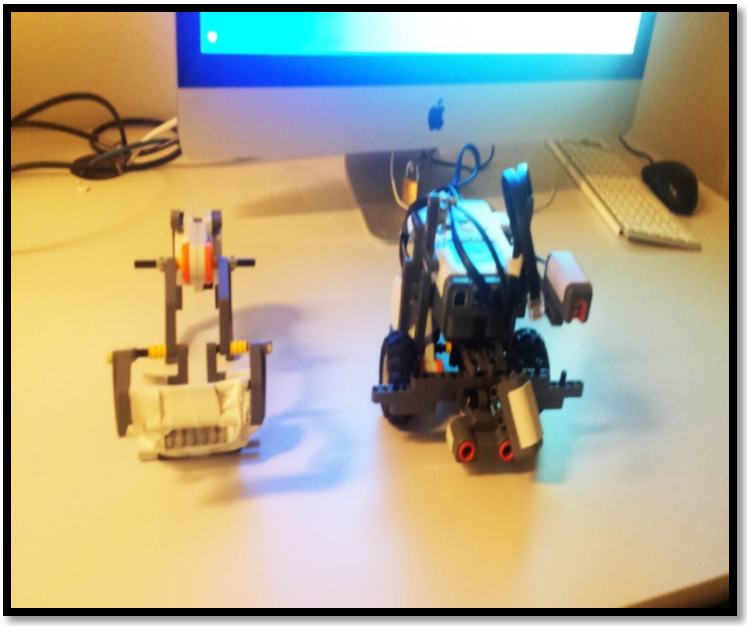
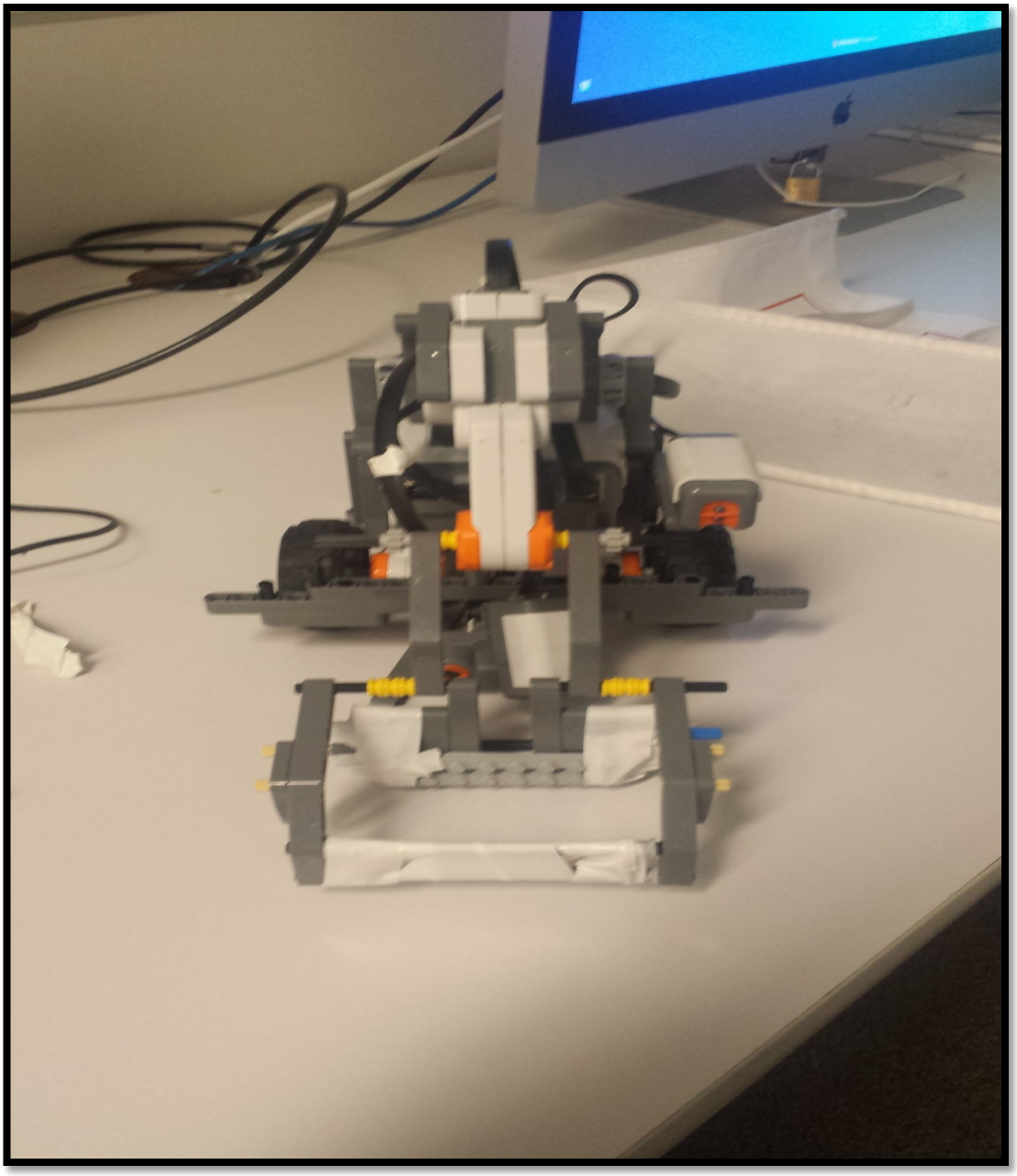
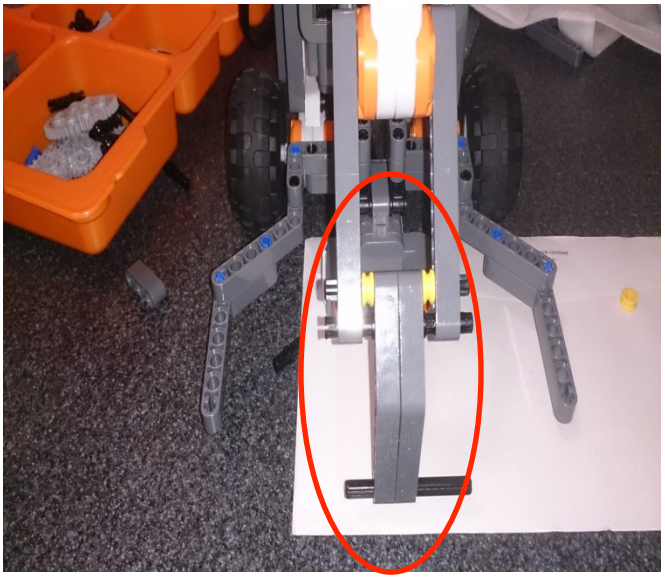


Image 36

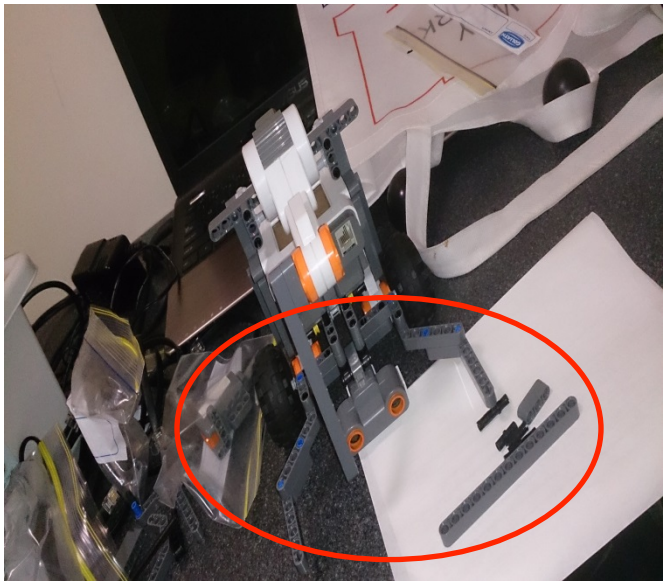


Changes made to design



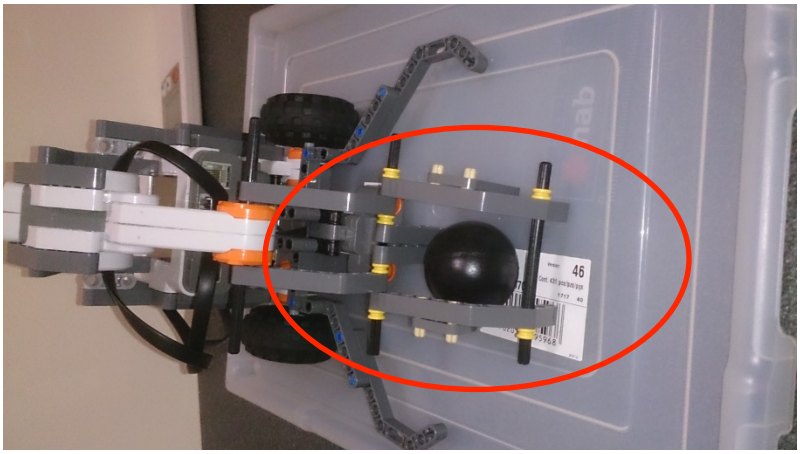
As shown in Image (a) our kicker was too wide and was blocking the ultrasonic sensor so we reduced the width

Image (a)



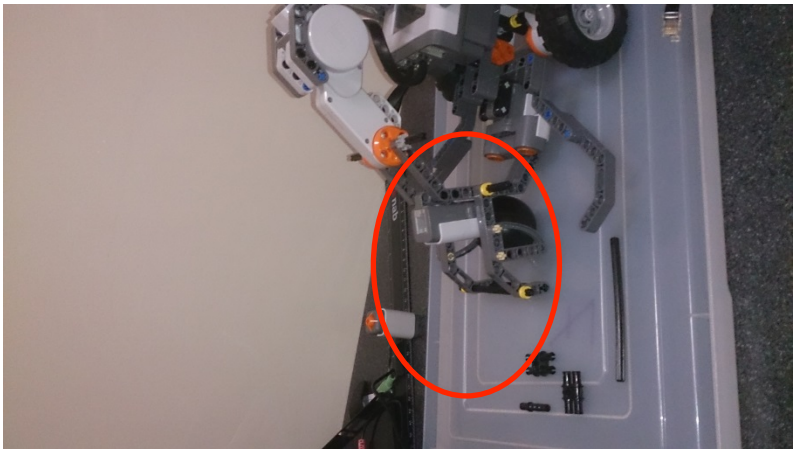
As shown in Image(b) we had a wide and long pouch, this was causing problems when trying to collect balls near a wall so it had to be removed

Image (b)



As show in image(c)Initially we had a less wider pouch which made it accurate for the penalty shoot task. But this proved troublesome while trying to capture balls during the field scan so it was made wider.

Image (c)



As shown in image(d) we initially had the light sensor in the upper part of the cage in order to make sure that when the cage was lowered a ball was definitely captured. But because we did not have a spare ultrasonic sensor the light sensor had to be used for the field scan as well hence its position needed to be changed

Image (d)

Programming Task 1- Penalty shootout

Our Robot uses the following approach to accomplish task 1:

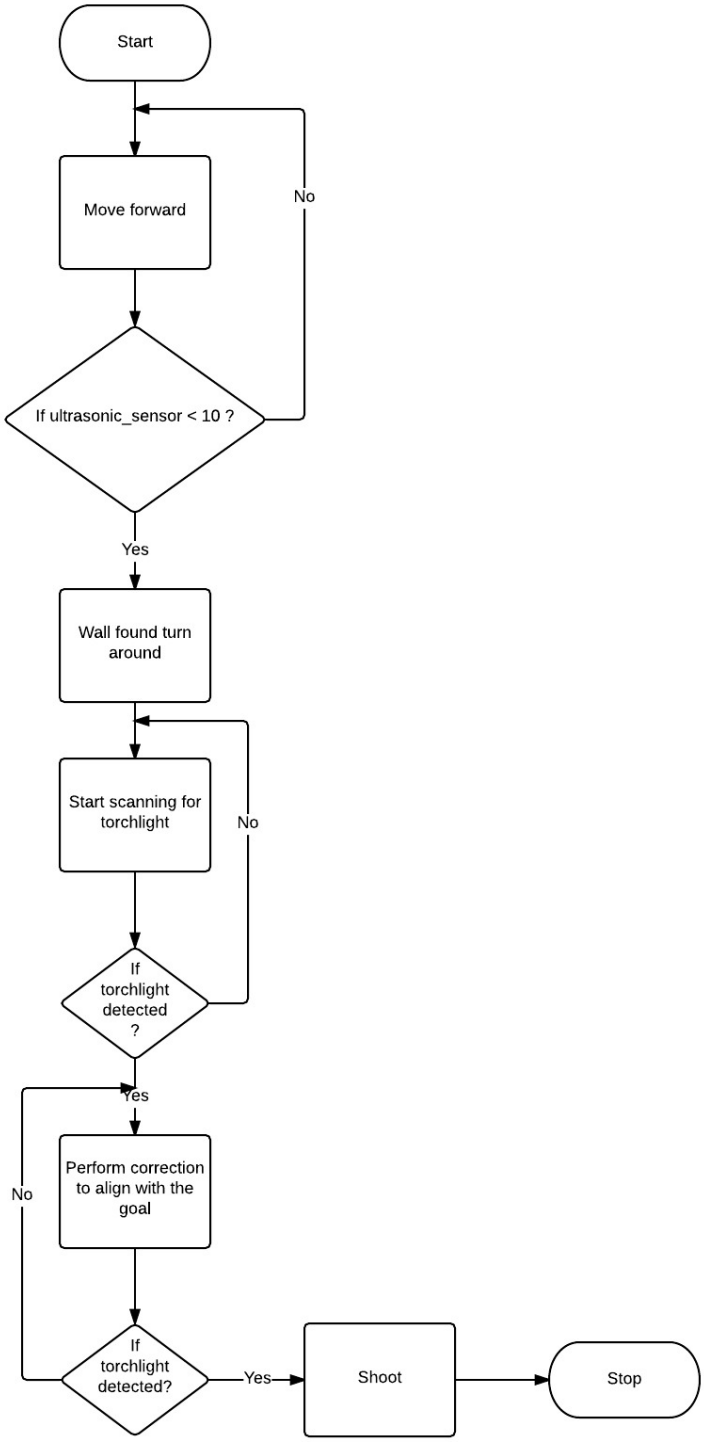


Figure: Block diagram – penalty shoot

Sensors declaration and Configuration:

NXC requires that you declare the ports and initialize the sensors before taking input from them.

```
#define US_SENSOR SensorUS(IN_3)

#define TORCH_SENSOR Sensor(IN_1)

#define BALL_SENSOR Sensor(IN_2)

//set sensors

SetSensorLight(IN_1);

SetSensorLight(IN_2);

SetSensorUltrasonic(IN_3);

//turn light sensor LED off

SetSensorType(IN_1,IN_TYPE_LIGHT_INACTIVE);

SetSensorMode(IN_1,IN_MODE_PCTFULLSCALE);

ResetSensor(IN_1);
```

STEP 1: Move forward towards the wall

The ball is placed in the pouch of the robot and made to move forward by giving power to the left and right motor.

```
while (true) {

        OnFwd(LEFT_MOTOR, 25);

         OnFwd(RIGHT_MOTOR, 25);
```

STEP 2 and 3: Check if the wall is near and Turn around

The ultrasonic sensor is used to detect and keep track of the distance of the robot from the wall. The robot keeps moving forward until it is more than 10cm from the wall.

When the robot is 10cm from the wall it starts turning around

```
if (US_SENSOR < 20 &&
back_back_turn_taken==false){

        //do a 180 degree turn

        OnFwd(LEFT_MOTOR, 60);

        OnFwd(RIGHT_MOTOR, 10);

        Wait(3000);

        back_back_turn_taken = true;

        OnFwd(LEFT_MOTOR, 10);

        OnFwd(RIGHT_MOTOR,90);

        Wait(700);

        OnFwd(LEFT_MOTOR, 21);

        OnFwd(RIGHT_MOTOR,20);

        Wait(1250);

        }
```

STEP 4: Scan for torchlight and perform correction

Once the robot turns around it starts scanning for the torchlight in order to align itself to shoot the goal

```
if (TORCH_SENSOR > 40) {
//turn left
                OnFwd(LEFT_MOTOR, 30);
                OnFwd(RIGHT_MOTOR, 5);
               Wait(3000);
//turn right after 3 sec wait.
               OnFwd(LEFT_MOTOR, 5);
                OnFwd(RIGHT_MOTOR, 30);
//light signal will break the loop
               break;
          }
```

STEP 5: Wait for torchlight signal to shoot

The robot keeps performing the correction in the previous step until it sees the torchlight again which is the signal to shoot the ball.

```
if (TORCH_SENSOR > 40) {
          //upclose kick the ball
            RotateMotor(OUT_A, 95, +180);
           Wait(300);
          break;
}
```

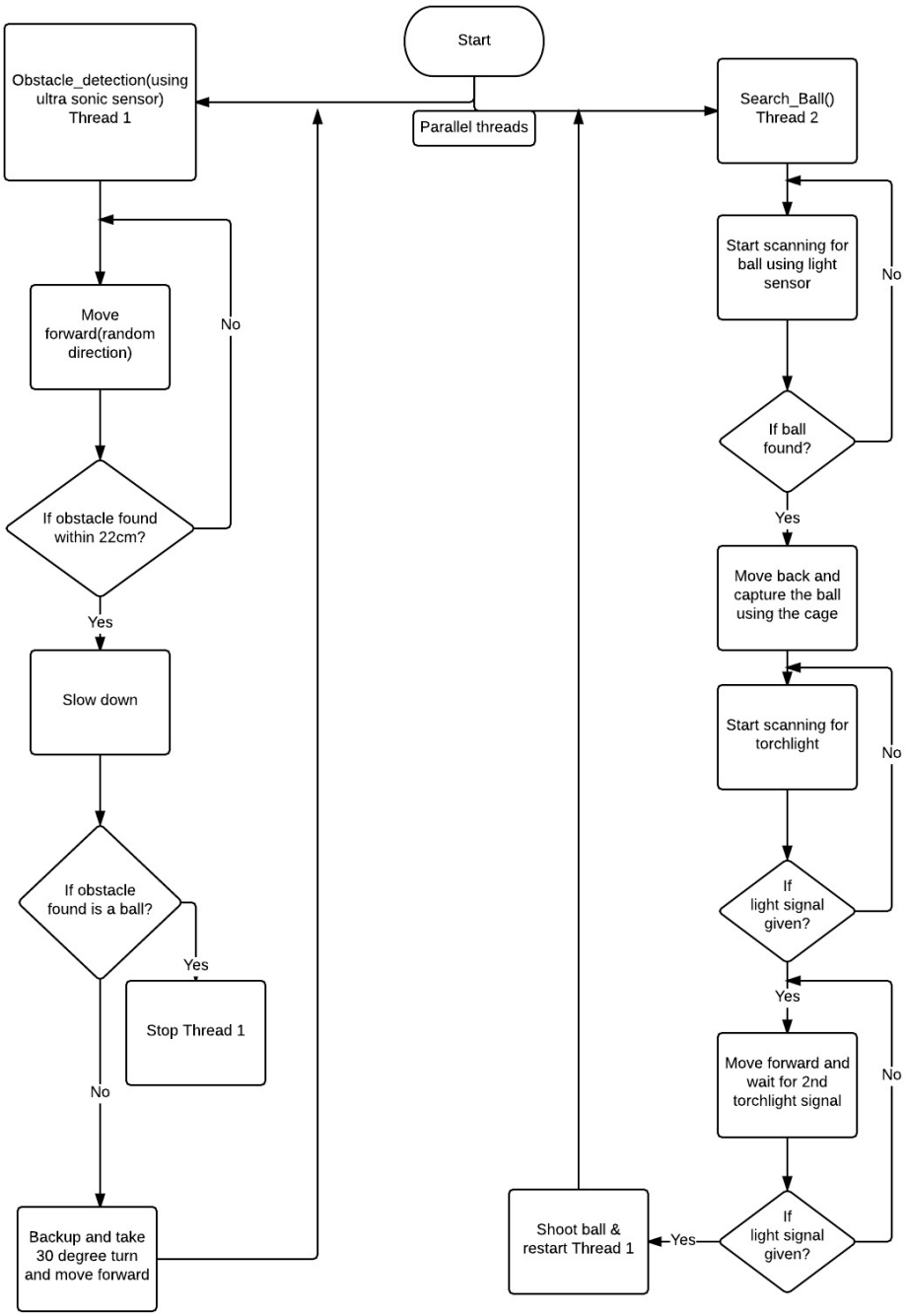


Figure 1.2 - Task 2 – Overview of Soccer Playing algorithm

Code-Listing 1.1: Thread 1(Obstacle Avoidance)

Obstacles Avoidance Function accomplished the following tasks:

1. Navigate around the field without hitting walls.

2. Slow down robot if a ball is found so that the robot can approach the ball and confirm it by light sensor.

```
task avoid_Obstacles(){
        int difference = 0;
        Wait(1500);
  while(true){
        if  (BALL_FOUND == false && CIRCLE_SCAN==true) {
                //rotate robot to initite the search
                OnFwd(LEFT_MOTOR, 50);
               OnFwd(RIGHT_MOTOR, 10);
                CURRENT_READING =  US_SENSOR;
                difference = PREVIOUS_READING - CURRENT_READING;
                //stop scanning if ball is found
                if (difference > 10) {
                        CIRCLE_SCAN=false;
                        break; }
                PREVIOUS_READING = CURRENT_READING;
        } if(BALL_FOUND == false) {
                if (US_SENSOR < 22) {
                        if (US_SENSOR < 7) {
                                OnRev(LEFT_MOTOR, 35);
                                OnRev(RIGHT_MOTOR, 35);
                        Wait(3000);
                        OnFwd(LEFT_MOTOR, 25);
                        OnFwd(RIGHT_MOTOR, 15);
```

Code-Listing 1.2: Thread 2(Obstacle Detection)

The previous function would slow down the robot If an obstacle is detected within 22cm. When the robot reaches within 10cm from the obstacle slow down further and slowly the robot will move forward to confirm if it's a ball. After the robot get a confirmation using a light sensor about the ball, it will quickly try to secure it using the cage. The following function will help accomplish this behavior.

```
task search_ball(){

while(true){

        if (BALL_SENSOR < 25){

                        //override the avoid wall function

                        BALL_FOUND = true;

                        Wait(200);

                        //back up a little

                        OnRev(LEFT_MOTOR, 26);

                       OnRev(RIGHT_MOTOR, 26);

                     Wait(500);

                        //put the kicker down

                     RotateMotor(OUT_A, 75, -180);

                              //Wait(300);

                             search_light();

                             approach_light();

                             OnRev(LEFT_MOTOR, 35);

                        OnRev(RIGHT_MOTOR, 15);


                        Wait(3500);

                        BALL_FOUND = false;

                        CIRCLE_SCAN = true;

                         }
```

Code-Listing 1.3 - Function (Search For Light Signal)

After the robot has secured the ball, robot will move in a circular manner and try to look for a light signal given using the torch.

```
sub search_light(){
        int counter = 0;
        int dia = 0;
        dia = 20;
        while (true) {
                TextOut(1,LCD_LINE1,"TORCH SENSOR");
                NumOut(2,LCD_LINE2,TORCH_SENSOR);
                if (TORCH_SENSOR > 40 || or dia>30) || counter>15 {
                        break;
                }
                //rotate robot to initite the search
                OnFwd(LEFT_MOTOR, 35);
                OnFwd(RIGHT_MOTOR, 5);
                Wait(100);
                dia = dia + 1;
                counter = counter + 1;
  }
}
```

Code-Listing 1.4 - Function (Approach light and shoot function)

After robot receives the signal it will then move towards the goal until another signal is given, indicating the robot to shoot the ball.

```
sub approach_light(){
        OnFwd(LEFT_MOTOR, 25);
        OnFwd(RIGHT_MOTOR, 25);
        Wait(1000);
        while(true){
                OnFwd(LEFT_MOTOR, 25);
        OnFwd(RIGHT_MOTOR, 25);
        Wait(1500);
        if (TORCH_SENSOR > 40){
                //upclose release ball and kick
                RotateMotor(OUT_A, 95, +180);
                Wait(300);
                break;
                }
  }
}
```

Challenges:

1) Localization: - Determining the position of the robot on the field would be required to truly make the robot autonomous and to perform a smart search function rather than going in a random direction to look for the ball. Our robot had no idea of its location on the field. A complex algorithm would be required to process and filter the environment in order to determine its current position. The robot would have to remember its previous positions and keep track using some kind of grid system that uses distances from the different walls and the goal to determine its position on the field. We decided that there was not enough time to test and implement such an algorithm and went with a random search function instead. Even though the robot could not be made aware of its position on the field, a torch could be used to guide the robot towards the goal.

2) Shortage and inaccuracy of sensors: - Most of the teams used two ultrasonic sensors which made the task of detecting balls and collision avoidance with the wall much easier. Sadly there were no more spare ultrasonic sensors for us to borrow and the other teams could not spare them. All that needed to be done would be to use one ultrasonic sensor located at the base of the robot to locate the ball and the second ultrasonic sensor higher up keeping track of the wall. This would be a very easy way to know if your sensor has seen a ball rather than the wall. But if you have only one ultrasonic sensor this makes it hard to make out if you have seen a wall or a ball as an ultrasonic sensor cannot distinguish between the two. We had to work with a single ultrasonic sensor and try to compensate for the second one using a light sensor. We initially tried to use the ultrasonic sensor to perform a circular scan similar to that of radar in order to determine the location of the ball. Once the ball was located we approached it and verified if it was a ball that was detected using a light sensor. This approach was working for a while but for some reason could not detect the balls accurately all the time and sometimes doesn't detect the ball at all. We are not exactly sure as to why these exceptions; we assume that because the scan worked using a circular motion maybe the ultrasonic sensor did not receive the sound signal back due to the rotation of the robot. Theoretically this should not occur but that is one of the main problems we face when programming for robotics, "expect the unexpected".

Performance of Robot

The robot was initially scoring goals accurately. We had made a change to the size of the cage and kicker. The cage was made wider for the sake of task two to increase the probability of capturing the ball. This change greatly affected the accuracy of the robot as the ball would angle away from its path depending on which side of the cage it was positioned at. The shot was most accurate when the ball was positioned at the center of the cage but this did not occur very often when the robot did the 180 degree turn. So the robot did not perform as well as we had hoped it to. By the time we realized the flaw of increasing the width of the cage it was already presentation time. We experienced a similar difficulty with task 2. Our initial approach was to use the ultrasonic sensor to perform a radar like scan to find the

ball and then use the light sensor to verify if it was ball or a wall. But this did not work out well as the ultrasonic sensor did not detect the ball most of the time. And using the same ultrasonic sensor to detect balls as well avoid collision with walls was not very practical.