Furkan Enes Yalçın Nazmican Çalık

CmpE 434 Introduction to Robotics Term Project:

NBA EV3

Introduction

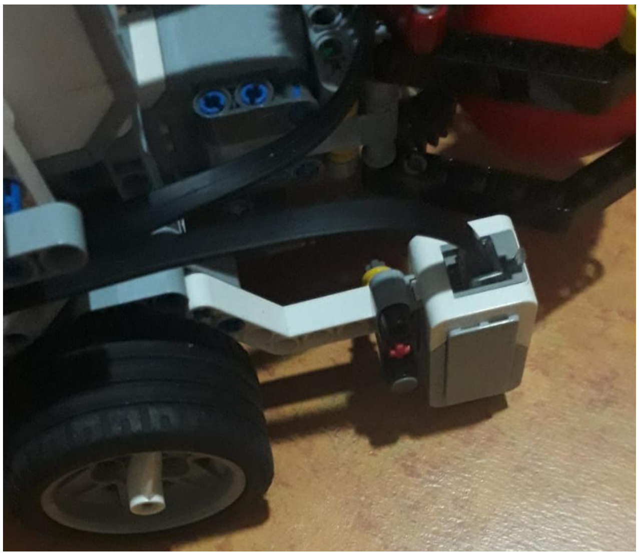
In this project we are asked to implement a scenario about a robot that has to carry balls and put them in a basket.

We are given with a map in this project. First we need to make a map of the environment and then make the robot grab the ball and then according to the map we will put it into some part of the map. We need to score 2 points for that purpose (to get out of the jail).

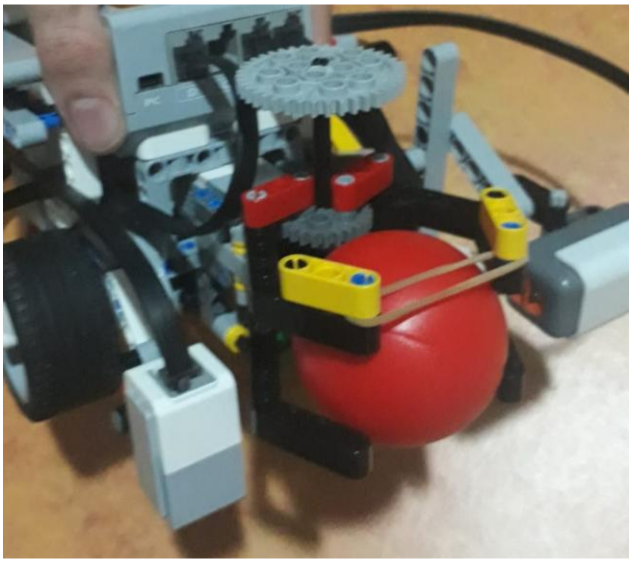
We need to implement two different tasks to accomplish above objects. We need to implement an idle state too. And since we may need to reset these tasks while they are being executed, we need to implement this function too. And we need to trigger these tasks using the button on the robot.

Mechanical Design

We decided to continue with our two wheeled robot with one additional ball caster.

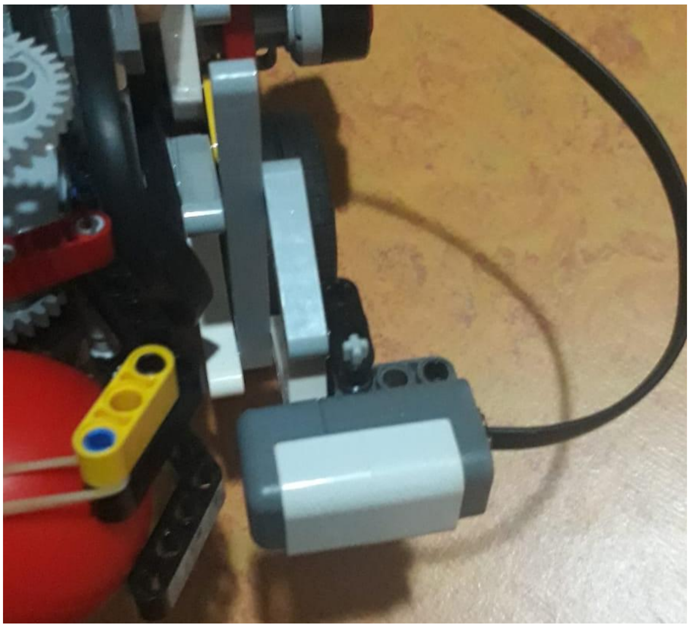


Since we need to determine what color the ground is; we put EV3 Color Sensor at the front right of the robot, facing to the ground.



We made a "claw" for picking up the ball. We looked at some design manuals for EV3 Mindstorm sets and get the design from there. We used the medium regulated motor for this claw. We put the claw on the front of the robot. We enforced this claw with a rubber band in order to make it more intact when it holds the ball.

We used one ultrasonic sensor and put it on the left of the robot. We think that with rotating the robot by 360 degrees and scanning the area with the ultrasonic sensor, we can draw the map of the area. Of course, with the help of the color sensor which faces towards the ground.



We are using an NXT Light Sensor on the front left of the robot. With its Red Mode, we can detect if the ball is blue or red. Its connection is a little bit loose because we may want to change where it is looking at.

We are not using gyro because we think that we can stay on course with the odometry alone. However, we can add the gyro if we will encounter some problems. We tried to keep our robot as small as possible in order to make it stay in a square of 33m2.

Software Design and Plan

We needed to implement two different software: one is running in the robot and the other one is running on the computer to show the map.

Since we needed a connection between the two codes we implemented a socket connection (actually we have used the previously implemented version)

We needed to implement for tasks:

* Map Making

* Localization

* Grabbing the ball

* Path Planning

We will try to explain what we have done or what we have planned for the project implementation.

Furkan Enes Yalçın Nazmican Çalık

We are trying to improve our design so, this document could be outdated when we are making the final demo.

1) Map Making:

For map making we are planning to use a data structure to hold the map data. Which can be an double dimension array or a graph.

If we make it a 2d array then we still have to implement a new data structure to hold the walls' locations. In order to be sure that we are in a given block we will use light sensor that is rotated towards the ground. We know that there are cells that are in different colors. If we are in a green cell for example we will know that we will get the ball from that cell (in the middle). So we put an enum on that cell (could be a simple integer) to determine what is there. For ball: 1, For white: 0, For hole: 2, For target: 3

2) Localization:

We know that we have to localize ourselves after we get the map of the environment. And according to our location then we will apply some search algorithm to find the balls position. Then we will head our robot there. It should be possible to detect where we are if we take look at our surrounding cells. We can do that by going to the neighbor cells. Since we know that the cells' dimensions are predetermined we can easily use the odometry data to to change our robots location.

3) Path Planning:

We have to implement a search algorithm here because we need to take some actions to get to the correct cell to get the ball. We already have the map of the environment so we can just apply the search algorithm and find the best route to our robot. We know that the ball will be in the middle of the cell so we can just get to the balls position.

4) Claw:

In order to get the ball we needed to use the medium regulated motor. We have predefined some angles to grab the ball. We made the robot go the half length of a cell and then open its claw using medium regulated motor with some predefined degree. After grabbing the object we rotated the medium regulated motor to the opposite direction with the same amount so we grabbed the ball. In order to get the correct angle we have tried a few times. We have used mostly constants for already predetermined degrees.

For now we don't have any general architecture but we are planning to use a convertible one. We will wait for the inputs in a big loop and then make the individual tasks different methods so that we can call them. We will call methods accordingly to the given inputs. Maybe if we get stuck with the architecture we can convert to subsumption architecture and make the tasks behaviors but that is not for sure for today.