Content:

1)Introduction

Our project is the robot that serves the purpose of picking up and segregating lego balls by their colors. Robot operates on a lego motor nxt lego brick, and the software is nxt brick software using robotc (Windows).

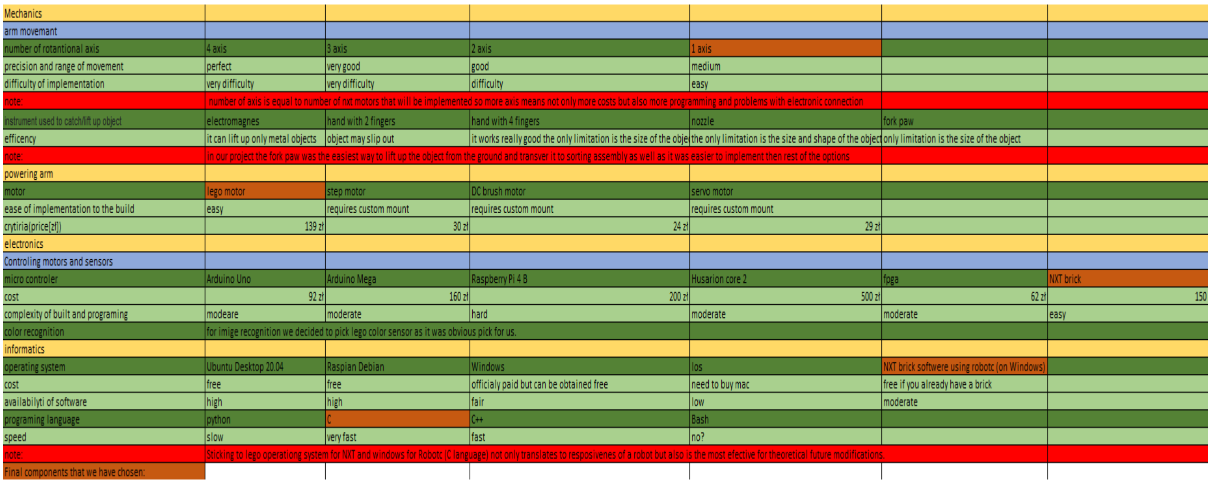
2)Purpose and scope of the work

The purpose of our project is to create a robotic arm capable of picking up spheres from the ground and segregating them based on their colour.

3) Overview of existing solution:

A hand for picking up the balls, pressurized nozzle for sucking the balls in. For sorting the balls a rotary system of containers can be used to make sure the ball falls in the correct container. We can also use a stationary array of containers and a guide rod to position the ball in the right container.

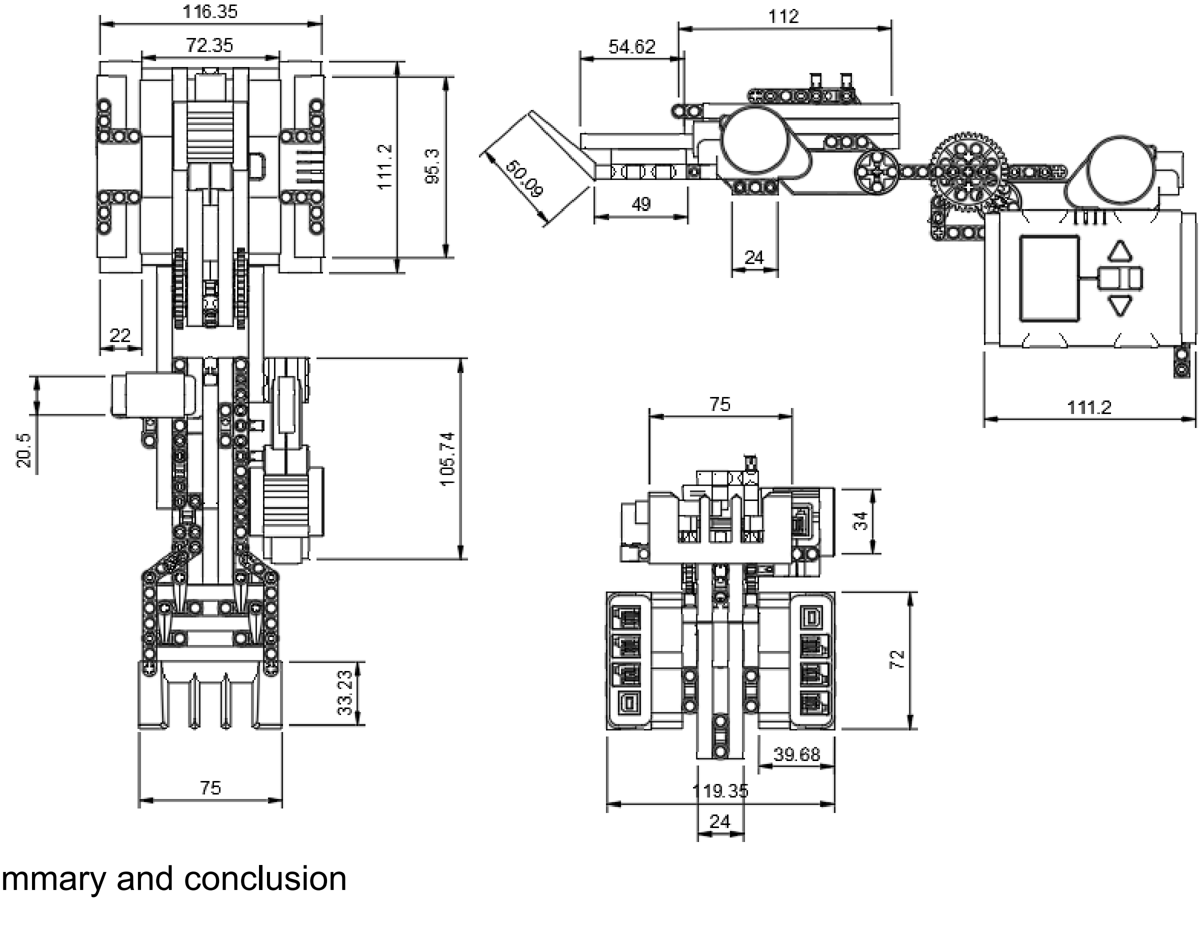
4) Morphological table:



Mechanics						
arm movemant						
number of rotantional axis	4 axis	3 axis	2 axis	1 axis		
precision and range of movement	perfect	very good	good	medium		
difficulty of implementation	very difficulty	very difficulty	difficulty	easy		
note:	number of axis is equal to number of nxt motors that will be implemented so more axis means not only more costs but also more programming and problems with electronic connection					
instrument used to catch/lift up object	electromagnes	hand with 2 fingers	hand with 4 fingers	nozzle	fork paw	
efficency	it can lift up only metal objects	object may slip out	it works really good the only limitation is the size of the obje	the only limitation is the size and shape of the object	only limitation is the size of the object	
note:	in our project the fork paw was the easiest way to lift up the object from the ground and transver it to sorting assembly as well as it was easier to implement then rest of the options					
powering arm						
motor	lego motor	step motor	DC brush motor	servo motor		
ease of implementation to the build	easy	requires custom mount	requires custom mount	requires custom mount		
crytiria(price[zł])	139 zł	30 zł	24 zł	29 zł		
electronics						
Controling motors and sensors						
micro controler	Arduino Uno	Arduino Mega	Raspberry Pi 4 B	Husarion core 2	fpga	NXT brick
cost	92 zł	160 zł	200 zł	500 zł	62 zł	150
complexity of built and programing	modeare	moderate	hard	moderate	moderate	easy
color recognition	for imige recognition we decided to pick lego color sensor as it was obvious pick for us.					
informatics						
operating system	Ubuntu Desktop 20.04	Raspian Debian	Windows	Ios	NXT brick softwere using robotc (on Windows)	
cost	free	free	officialy paid but can be obtained free	need to buy mac	free if you already have a brick	
availabilyti of software	high	high	fair	low	moderate	
programing language	python	C	C++	Bash		
speed	slow	very fast	fast	no?		
note:	Sticking to lego operationg system for NXT and windows for Robotc (C language) not only translates to resposivenes of a robot but also is the most efective for theoretical future modifications.					
Final components that we have chosen:						

The first problem is how many rotational axes we will use. Our biggest concern was how many nxt motors will be used as our possibilities in terms of connecting them to our future microcontroller were limited to maximum 3 where we needed to connect our sensor too. one axis appeared to be not that bad solution because even if its precision isn't so good it is solving most of our problems and our arm is not losing too much in terms of effectiveness.This solution is really light in comparison to other options. Because we have chosen 1 axis arm the best choice in terms of instrument used to lift up was fork paw as it is the most efficient working tool in such a set up plus it is not using any additional motors or electronics. In terms of choosing our motor we decided to go for the one that is the easiest to implement in our build so the lego motor was the best choice. In picking operating system we really just wanted to get something easy when it comes to programming and something not so complex in terms of build and after some discussion and we came to conclusion that optimal choice will be the nxt brick with free and moderated software that is nxt brick software. The biggest problem in the programming section was to make such a code that will make moves of our robot fast so our programming language should be fast as well so we chose language C.

6) measurements:



8) Summary and conclusion

From the beginning our aim was to design a robot capable of recognizing a ball, autonomously driving towards it, picking it up and sorting it into a right container. The module capable of doing that would be called an arm.

The arm would consist of three stepper [Go1] motors and would be connected to an Arduino via a L298N H-Bridge Motor Drive (one per two motors). The arm would pick up objects using a four fingered hand driven by one stepper motor hence the necessity for two H-Bridges since the total motor count in the arm is four.

The problem with this solution was that the stepper motors on the market that were in our budget had negative characteristics like high weight to torque ratio and had suboptimal mounting points. They would require additional 3d printed parts to be able to be mounted which again added weight. Due to the fact that the motors were mounted on the arm it was crucial for us to minimize weight on the arm to increase its carry potential.

To solve this problem we decided to use Lego NXT motors due their ease of implementation into the design already built using Lego technics components. The use of Lego NXT motors would actually add to the structural integrity of the design due to their robust build and three four-holed mounting rails for Lego technics components. We wanted to connect the Lego NXT motors to the same L298N H-Bridge Motor Drive by simply stripping the proprietary Lego NXT Mindstorms connectors and soldering them to the wires that came with the Arduino starter kit. The wires then could be connected to the H-Bridge.

We were originally going to use an Xbox 360 Kinect with ROS to determine the ball’s position in 3D mapped space by using an array of one infrared camera, one ultrasonic sensor and two cameras. This solution would give us precise coordinates of the ball. Unfortunately, the computer that was available to us, a Raspberry pi 4B 4GB ram was not powerful enough to handle such calculations. The mapping, processing and image recognition of one frame rarely took less than two seconds and the device that was passively cooled by a small copper heatsink quickly thermal throttled. As the robot travelled towards the object the unknown variables linked to the movement of the robot through space made the initial map not precise enough to accurately drive to the ball and pick it up. If we had more processing power, we could have done the mapping and object positioning once or twice a second which would probably give us real-time precise position of the ball. Unfortunately we did not have access to such a device.

To solve the problem of getting precise real-time position of the ball we moved to a remote-control base which allowed us to finely control the robot using our brain's computing power. As for the hand, we decided to design a one piece sorting assembly which at its end had a 3D printed attachment that allowed us to pick up the balls from the ground when the assembly was in its zero-position. After getting hold of the ball the arm lifts up and uses gravity to force the ball down to a compartment where the ball is stopped by a rotating gate and analyzed by using a Lego RGB sensor. Having read the ball's color, the sorting assembly rotates a four compartment cylinder aligning the proper cylinder with a funnel connected to the aforementioned sensor compartment. The gate rotates releasing the ball, which falls through the funnel into the compartment, consequently being sorted. The assembly is positioned downwards to the ground in its stand-by mode. When activated by a button press it automatically raises, reads the ball's color, selects the right container and rotates the gate sorting the ball. When all balls are sorted, the arm returns to its original position.

This design is much faster and efficient then the previous iterations. It is much more reliable than the previous design and uses only three motors, one for moving the arm, one for rotating the gate from the sensor compartment and last one for selecting the sorting cylinder.

Given that we have streamlined our design down to three motors we are now able to use the original Lego NXT Mindstorms brick which has connections for three Lego NXT Motors and three Lego NXT Sensors. Using the Lego NXT Mindstorms brick has many advantages. It is compact compared to an assembly of Arduino and L298N H-Bridge Motor Drives, it is self powered by using AA batteries and it allows us to use the robotc environment which let us quickly develop software for it.