

INTEGRATING PROJECT - Assignment 1 write up

by Morgan Jones

FILE	3676_MORGAN_JONES_INTEGRATING_PROJECT_-_ASSIGNMENT_1_WRITE_UP_93631_2118573587.PDF		
TIME SUBMITTED	30-APR-2018 03:31PM (UTC+0100)	WORD COUNT	1858
SUBMISSION ID	87914767	CHARACTER COUNT	7803

INTEGRATING PROJECT

Assignment 1

Lego Mindstorms maze

Introduction to Project

As part of this Module I have been set an assignment containing a problem that requires solving. The point and aim of this assignment is to construct a Lego Mindstorms robot working as a team while solving specific problems along the way. The problem all teams have been set requires the construction and programming of a robot using the Lego Mindstorms EV3 programming software. This software will allow us to be able to manually navigate the robot using a selection of programming tools. These tools allow us to make sure the Riley Rover robot can move from a set start point to a finishing point using the skills that we have been taught in class. The problem contains the robot follow the line, negotiate the block, re-join the line, enter and negotiate the maze, and finally reach the finish point. At this point, the robot must stop and indicate it has reached the finish point. This can be done with a tune.

Background Research

Before completing the project, I have spent time researching the Lego Mindstorms software to find out exactly what I will need to know. The software has helped me learn a lot about programming. This will be done using the types of programming that we applied while building our robots. I also have experience with using Lego mindstorm that to my university classes

Project Plan

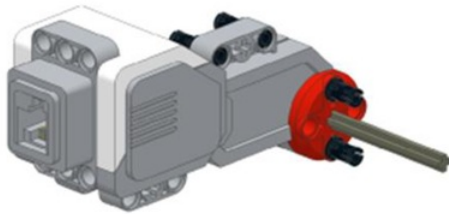
The Plan of our Integrated Project is that I (Morgan) will be creating and design the robot that will go around the maze. I will also be creating the code for the robot to circumvent the object blocking its path it will be done by using the ultrasonic sensor I will also need to use timer as once the robot has passed the blocking object it must return to its original path. Matt (Matt Howells) will be working on the code for the robot once it has passed the obstacle so that it will head towards the start of the maze and finally Bart will work on creating the code so that the robot can navigate the maze without crashing it will also being a colour sensor as the maze will have black walls

Design Process for the Robot

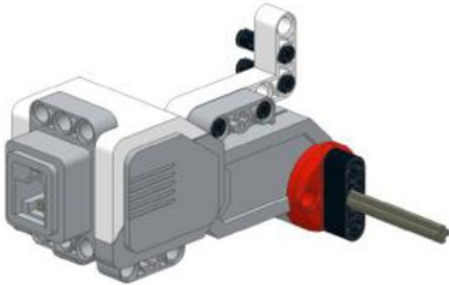
The robot that will be used is called the Riley Rover

First I Assembled the Right Motor

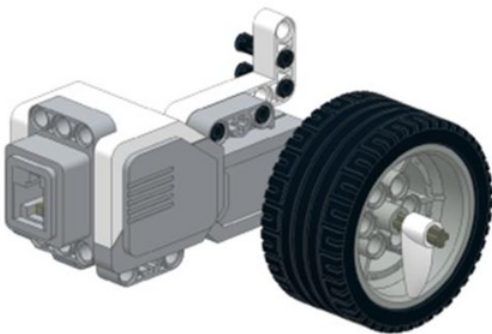
I added 4 black connectors and a rod to the motor



I then connected a block to two of the connectors to stop the wheel from getting too close to the spinner I then connected an L shaped block with 2 black connectors on it.



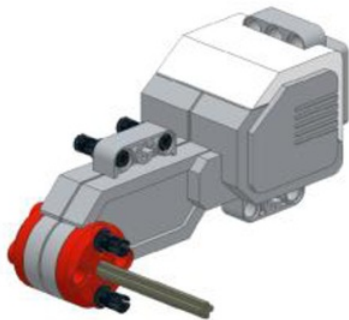
Finally I connected the wheel and added a fin to be used as stop so that the wheel does not slip



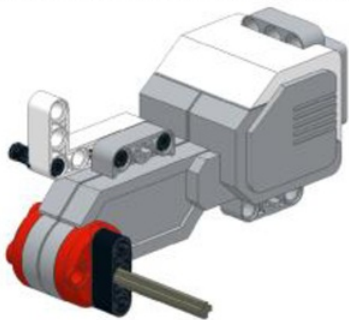
for the Riley Tachyon EV3 - www.rileyrover.com

Secondly I Assembled the Left Motor

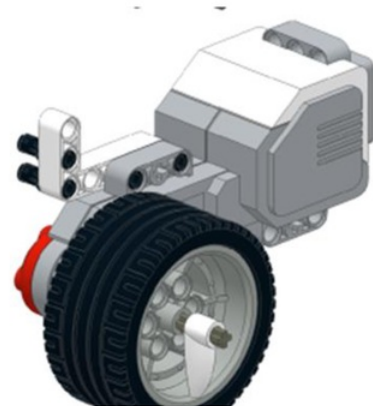
I added 4 black connectors and a rod to the motor



I then connected a block to two of the connectors to stop the wheel from getting too close to the spinner I then connected an L shaped block with 2 black connectors on it.



Finally I connected the wheel and added a fin to be used as stop so that the wheel does not slip

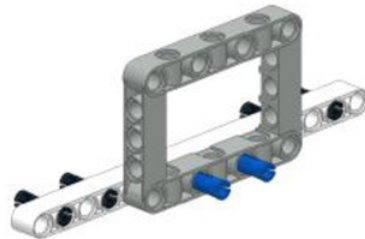


Thirdly I assembled the back castors

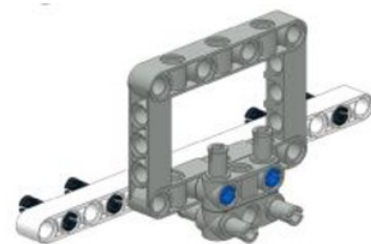
I connected 2 long blue connector on to a long white block I then added 4 small black connectors to the opposite side



Then I added a grey window on to the 2 blue connectors



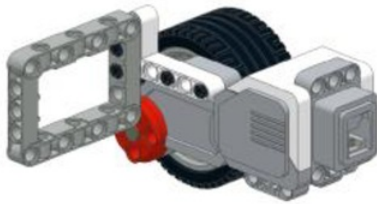
I also added 1 grey double edged connector to the blue connectors the added another grey double edged connector and connected it the first grey connector



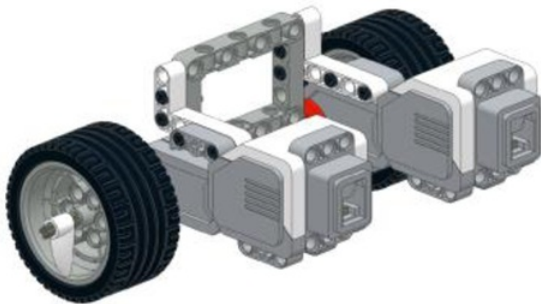
Lastly I connected the a sliver ball to act as a wheel of sorts as it spins in all direction which helps with turning



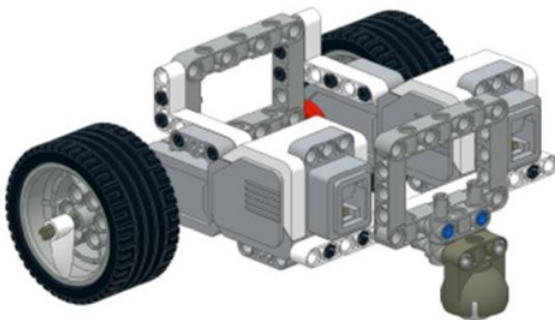
Fourthly I assembled the motors on to the back castors
I connected a grey window on to the right motor via 2 black connectors



I did the same with the left motor making them 1 object instead of 3 separate ones

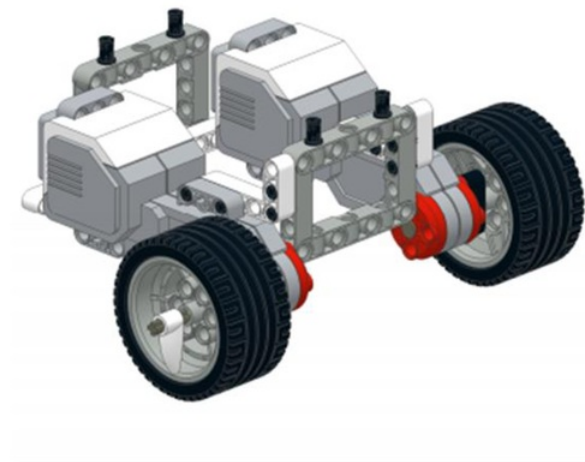


I then connected the back castor I made earlier using the same method as just now but using 4 black connectors instead of 2



Lastly Attach Power pack – “Brick”

Lastly I attached the power pack to the body of the robot via 4 black connectors in to the bottom of the power pack



Finished Design without sensors



Connection of sensors

I connected a colour sensor on the front facing down so that it can follow the black line it is connected to a power pack via a ether cable I also connected an ultra-sonic sensor to scan for walls or any other obstacles in the way so that the robot knows when to turn this is also connected via ether cable



This is the original final design



The final design

Minor changes from the one above with the addition of the extra ultra-sonic sensor to help with navigating around the maze the





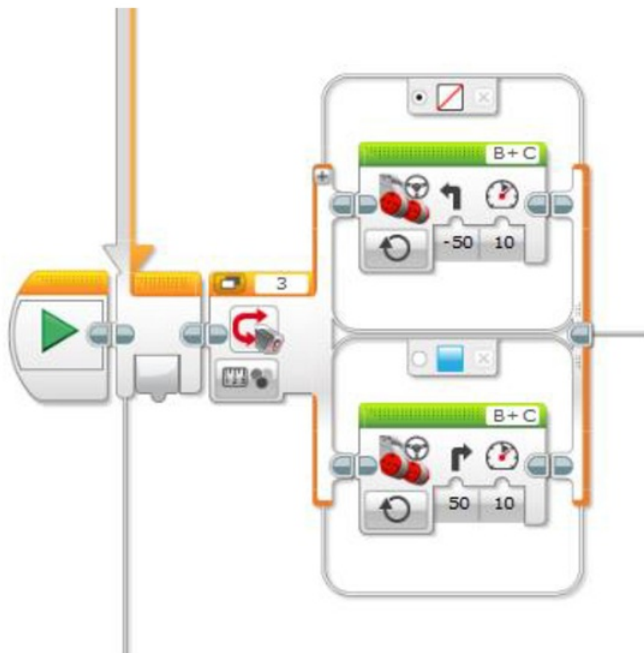


Design Process for the avoiding the obstacle

First the design the code for the colour sensor so that I will follow the line set out for and change direction if the robot strays from the line for example the line is white and there is a blue line on one side of it and a red on another side if the robot goes on blue then it turns right and if on a red it turns left.

What the code does when running.

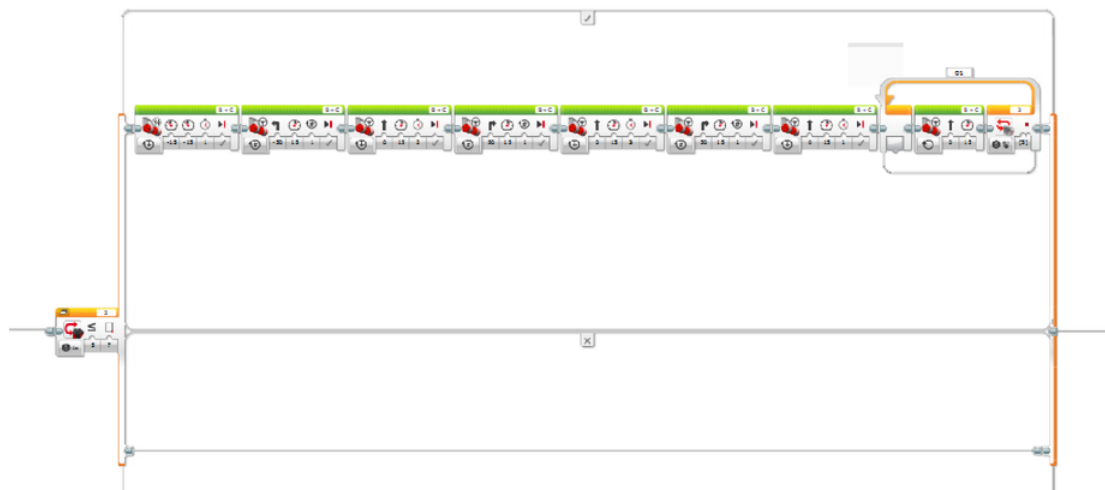
The colour sensor is in a switch as there is two required settings and here they are on any colour the robot turns right but if the robot strays on to any colour not blue then it will turn left so instead of following a straight-line pattern you will have a zigzag pattern meaning that as the robot follows the set path it will slowly zig zag straight ahead then it will do a sharp turn to the left where it will continue to go straight ahead then it will take a sharp right where until it reaches the obstruction where the next part of the code will start working this means that the task for which the code was made for was completed without any major issues.



Secondly, the design code for the ultra-sonic sensor, which will scan for any object in front of the robot then the code will tell the robot what actions it will take, whether it will turn left or right, how long it goes in that direction and finally turn left or right once it has reached the edge of the obstruction and how long it goes in that direction before it finds the line again.

What the code does when running.

The robot will travel forwards while the ultra-sonic sensor scans for the obstacle. Once the obstacle is found, the robot will reverse backwards at a speed of minus fifteen for one second so that there is enough room for the robot to turn without crashing. The next step is for the robot to turn minus fifty to the right at a speed of fifteen for one rotation then it will go straight ahead at a speed of fifteen for two rotations. It will then turn fifty to the left at a speed of fifteen for one rotation it will then go forward at a speed of fifteen for three seconds then it will turn fifty to the left at a speed of fifteen for one rotation then it will straight ahead at a speed of fifteen for one second. Finally, the robot should have reached the line and started re-following it by using the colour sensor at a speed of fifteen then switch to another piece of code as the maze has no set path for the robot to follow meaning that the task for which the code was made for was completed without any major issues.

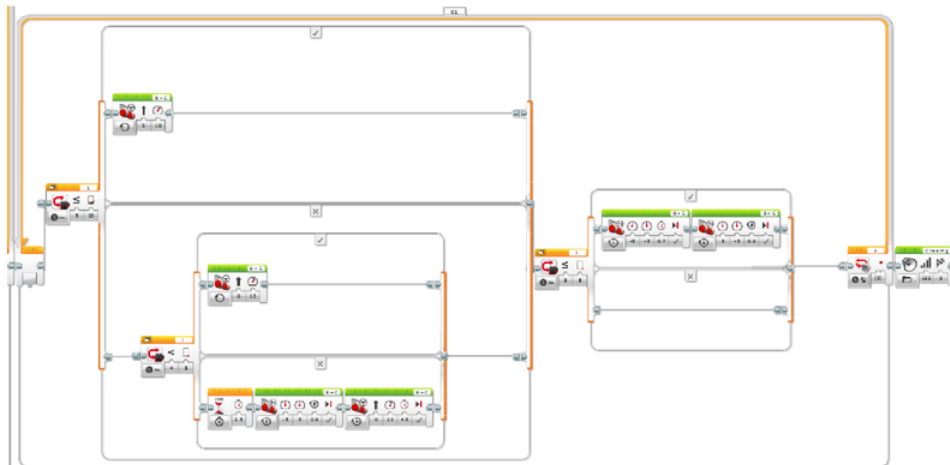


Design process for the maze

The robot will follow the right wall so the robot for example will enter the maze and keep track of the right wall via an ultra-sonic sensor positioned on the side of the robot so if the sensor the robot will turn right following the wall but if there is a wall ahead and to the right then the sensor on the front and side of the robot will pick them up making the robot turn left to avoid hitting the maze wall and so that it will keep following the right wall the this code will repeat loop multiple times but it does depend on the design of the maze

What the code does when running

The robot will enter the maze and hug the right wall so that the ultra-sonic can keep track of where the wall is once the ultra-sonic sensor can no longer scan the wall the robot will turn right so it can keep following the wall. The code will tell the robot to turn left as the front sensor as picked up the wall ahead and the side sensor is still picking up the wall on the right side then it will continue to go forwards until the front sensor picks up another wall in front of it making it turn left again then it will go straight ahead and turn right as the side sensor can no longer find the wall then it will go forwards until the robot has to turn left as the front sensor has picked up the wall ahead and the side sensor is still picking up the wall on the right side, it will turn left again and finally it will go straight ahead, it will stop at the finish line which will then cause the robot to play a tune letting you know that the robot has completed the maze and that it has stopped moving and meaning that the task for which the code was made for was completed without any major issues



Conclusion

The video shows the completed code in action as the robot follows the path set out for it without any issues. Once it reaches and scans the obstacle, it correctly reverses and circumvents the obstacle without knocking into it. The code also works correctly in reference to finding its path once it has bypassed the obstacle. The code that took the most work and was the most troublesome was the code to navigate the maze, as it was designed to work in any in maze no matter what it looks like so thankfully it works and does not crash into any walls in or around the maze causing the practical part of the assignment to be an overall success even if there were a few issues along the way

[Video of robot following completed code](#)

Special recognition for my partners Bartosz Galantowicz and Matt Howells in this assignment as their help and cooperation was invaluable

INTEGRATING PROJECT - Assignment 1 write up

GRADEMARK REPORT

FINAL GRADE

74/100

GENERAL COMMENTS

Instructor

Assessment Component 1 (Report):

Introduction: Quite well stated, a diagram would be useful here as this is a visual assignment. 4/5

Background Research: You say you have spent time researching, but what did you do? You need to detail this and provide references to show what inspired you to take the approach you did.

0/15

Project Plan: You need to make a list of tasks and time allocated, however, you have indicate who will be doing which task and the order they were done in. 5/10

Design Process: Very clearly shown for the robot. Don't forget to say *why* you adopted this approach - why is this design suitable? Nice screenshot with the final robot! Very good inclusion of the code for your section in particularly. Well done. 34/40

Evaluation and results: Although you have linked to the video, you need to show some screenshots here. I think you could be even more positive as this was a very good implementation. this section is a little short. 17/30

Assessment 1 Total:60 %

Assessment Component 2 (Demo):

Technical knowledge/skill 19/20

Analytical and Problem-solving skills 19/20

Quality of result 57/60

Assessment 2 Total: 95%

A really good implementation, just a tiny bit of adjustment needed to avoid hitting maze wall. Very good effort, transitions managed and definite end point. Well done.

OVERALL MARK FOR MODULE: 74%

PAGE 1
PAGE 2
PAGE 3
PAGE 4
PAGE 5
PAGE 6
PAGE 7
PAGE 8
PAGE 9
PAGE 10
PAGE 11
PAGE 12
PAGE 13
PAGE 14