USER MANUAL

FOR ROBOTIC CAR

Project 1



BY: PAULINA OSIKOYA,STUNDENT

Teacher: Emer Cahill

Table of contents:

Introduction

Checklist

Assembly of robotic car:

1.1 configuration of stripboard

1.2 Soldering

1.3 testing the stripboard

1.4 testing LEDS

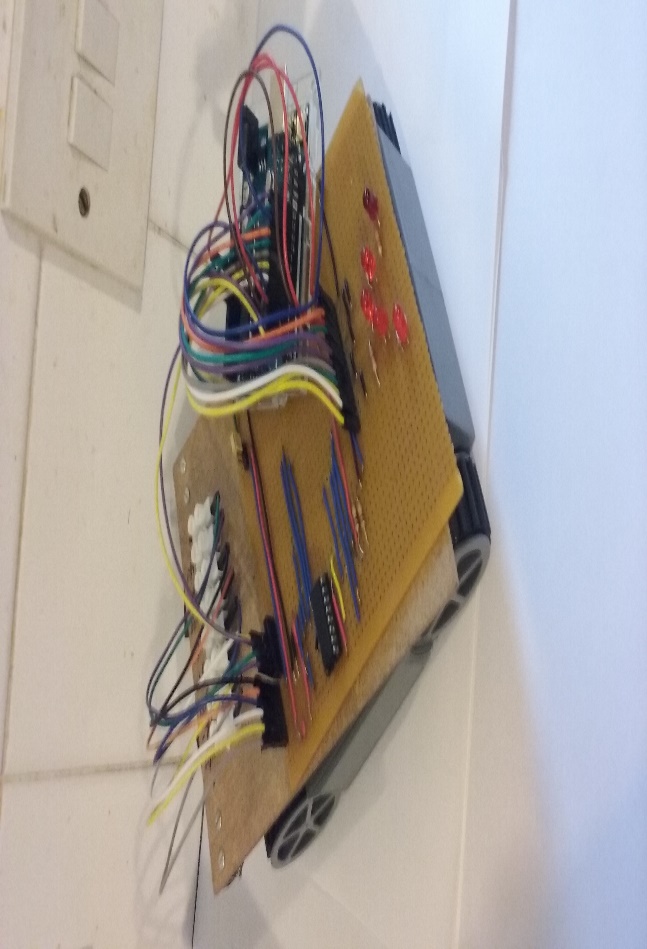
Programming the Arduino:

* 1. LEDS code
  2. Bumper code
  3. Making the robotic car drive
  4. Making the robotic car turn
  5. Controlling the speed of the robotic car
  6. The basic drill for the robotic car

Setting up the Robotic car

Implementation

**Introduction**

****

**This manual is designed and constructed to aid you in building your very own robotic car which, will be able to the following:**

**Your robotic car will be programmed to accelerate forwards, backwards, turn and be able to detect object/s and reverse from the object/s respectively.**

**Although the picture above of the finished project may look daunting, with these step by step instructions and in-depth explanations, be assured that you too will also capable replicating this project provided you adhere to this manual with attentivity.**

**Best of luck!**

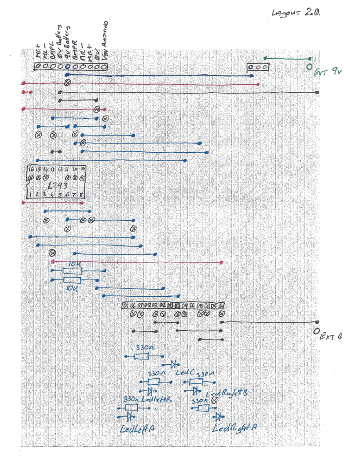
Checklist

In order to construct the robotic car, you will need to make sure you possess/ have the following to progress with your project:

* Tool kit which contains the appropriate equipment
* 6x 1.5v batteries
* A colour assortment of medium single stranded wires
* Solder iron and solder
* *10 pin socket*
* 16 pin chip carrier, into which the L293D chip will be inserted
* 1x L293D chip
* *14 pin socket*
* *3 pin strip*
* *2x Single pin*
* *Track breaker*
* *5x leds*
* *5x 330 ohm resistors*
* *2x 10K ohm resistors*
* *Arduino board and USB wire*
* *Arduino coding application*
* *Casket for the robotic car*
* Box to hold the condiments of the project
* 1x pair of crocodile clips
* A Multimeter

Assembly of Robotic car

Stripboard layout



Configuration of the stripboard

Objective: to successful construct the stripboard for the robotic car

The first thing you are going to construct in the initial stage of this project is the stripboard.

The stripboard is the most fundamental part of the project because it is from here that all the connection between the Arduino, the 14-pin socket, 10-way connector and the leds will take place. Therefore, it is crucial that special attention and time is given to stripboard premised on the fact that if the stripboard doesn’t function then the robotic car will in turn not function properly

1.1 Assembling the stripboard

Attached in the previous page is the strip broad with the appropriate connections and positions.

Your objective is to replicate this stripboards design perfectly because it is the basis of this that you will connect other objects later on i.e the Arduino boards.

Tip:

When configuring your stripboard, it is advised to take your time and start in a systematic and both sequential approach i.e. starting first with the wires first then the leds and lastly the sockets and connectors. It is also advised to not solder anything on initially as it is very common to mistakenly place a wire or socket in an incorrect position, so be sure to double check your wiring; making sure everything connects to one another and the positioning.

1.2 Soldering

Objective: in this section, you will acquire the skill of soldering

Before you begin soldering, it is important that you are aware of these two facts:

1. The soldering iron that you will be dealing with is very hot and it is important to exercise safety precaution like wearing protective material to prevent accidents from occurring
2. Once you solder something, it is a very tedious to undo, so it is recommended to solder something only once and correctly

Once you begin soldering make sure you have enough solder at hand and that your soldering is hot.

Once you have positioned everything on your stripboard, flip it on its other side and begin to solder the free wire in view (everything on the stripboard must be soldered on.)

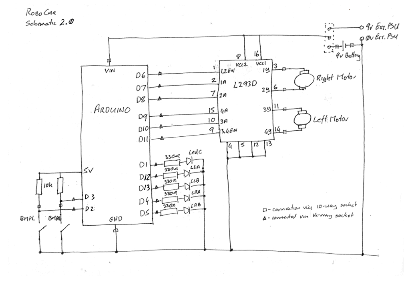
Solder the free wire by placing the soldering iron next to the track while adding the solder to the track so it melts and creates a cone-like mold.

Tip:

When you begin soldering hold down the solder iron down on the track on which you are soldering, this makes the track hot and the solder will melt quicker

Schematic diagram

The schematic diagram shows you how all the connections between the stripboard and the Arduino are derived.



1.3 Testing the stripboard

Objective: the purpose of this is to see if all the connections are in place and all the positions in place so there will be no errors in the functionality of the robotic car later down in the line which can be fix/ sighted now

In this section, the project you will need a Multimeter to do your testing!

Before you begin testing the strip board place the attached schematic in the page before and a layout diagram where you can see them.

Set your Multimeter to Digital Voltage Meter to 'beep' when shorted

Firstly, check the power supply and whether there is a shortage:

Make sure that **no beeps** occur, (i.e. there is not a connection), when one meter probe is placed on the Ext 0v pin and the other on any of the Red, Green, Purple or Pink wires.

This make sure that there are no shorts between any power lines and ground.

Here is a checklist of all the connections you should hear beeps for this means that the parts on the stripboard are connected and this is good:

* The 5v pin on the 14-way connector and each of the 10K resistors
* The 10K resistor and D2
* The other 10K resistor and D3
* D2 and BMPL on the 10-way connector
* D3 and BMPR on the 10-way connector
* 0v on the 10-way to the Ext 0v pin
* Vin Arduino to Pin 8 on the L293
* Vin Arduino to Pin 16 on the L293
* D6 to Pin 1 on the L293
* D7 to Pin 2 on the L293
* D8 to Pin 7 on the L293
* D9 to Pin 15 on the L293
* D10 to Pin 10 on the L293
* D11 to Pin 9 on the L293
* Pins 4, 5, 12 & 13 on the L293 to Ext 0v Pin
* L293 Pin 3 to MR+ of the 10-way connector
* L293 Pin 6 to MR- of the 10-way connector
* L293 Pin 11 to ML+ of the 10-way connector
* L293 Pin 14 to ML- of the 10-way connector

Tip:

It is best to go through the testing methodically and systematically to solve minor errors that could occur now

If it occurs that there is an error in the robotic car it is crucial to solve it now! Rather than later as this could hinder the robotic cat from functioning correctly

Minor things like forgetting to solder an item or mistakenly misplacing an item on the stripboard could result in going back to the section configuring the stripboard in the manual that is why it is very important to spot these issues now.

1.4 Testing the leds

Objective: to see if each of the individual leds are connected properly

In this section, you will need a power supply and crocodile clips to test your connections and you will also need your schematic to see connections and the stripboard layout page to see how all the parts are laid out

To see if the Leds are functioning set up a power Supply 5v D.C. power supply and connect the Ground of the Power Supply to the Ext 0v Pin on your stripboard with your crocodile clip connect 5v from the Power Supply to one end of a jumper wire using a crocodile clip.

it is the code on the Arduino that will cause the Leds to light so in order to start the Arduino you must send 5v to the appropriate pins to turn on the Leds

Here is a list of the relevant digital pins to light the Leds and instruction regarding the connections:

\*To know which Leds are which, refer to your strip board layout page for clarification\*

* Insert the other end of the jumper clip into D1 of the 14-way connector - LedC should light
* Move the 5v jumper into D12 of the 14-way connector - LedLeftA should light
* Move the 5v jumper into D13 of the 14-way connector - LedLeftB should light
* Move the 5v jumper into D4 of the 14-way connector - LedRightA should light
* Move the 5v jumper into D5 of the 14-way connector - LedRightB should light

Tip:

If it occurs that the leds are not flashing/illuminating then it would be advisable to check whether the resistors and leds are connected properly

It is also recommended to check if the proper voltage from the power supply is being sent to the leds

**Programming the Arduino**

Programming on the Arduino

You will be using the application arduino to code all of your codes as you it is the arduino board that is utilized for the robotic

1.1 LEDS code

Objective: this code is designed for the purpose of testing the leds making them illuminate and dim at set times

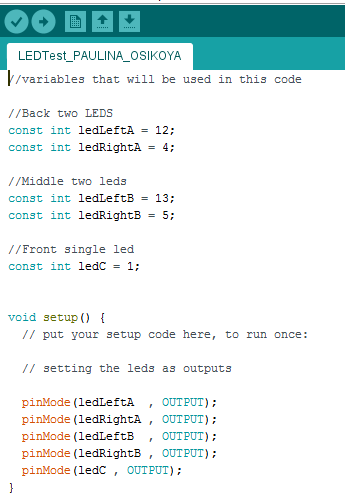
Method:

* Open a new file in Arduino
  + In the first section of Arduino before the void setup you will use 'const int' statements to assign names to all the LED pins. Use short names that mean something, for example name the Leds, ledRightA and LedLeftB and so on

* + Then in setup, declare all the LED pins as output

* + in the loop section of Arduino, you will write a code the will illuminate the two back Leds on and then off and the middle Leds on and off and then the single front Led on and off. This will in turn create a ripple of light among the Leds

The code should end up looking something like this:





1.2 Bumper code

Objective: this code is designed to test the bumpers for detection of touch and the leds will correspond to the bumpers

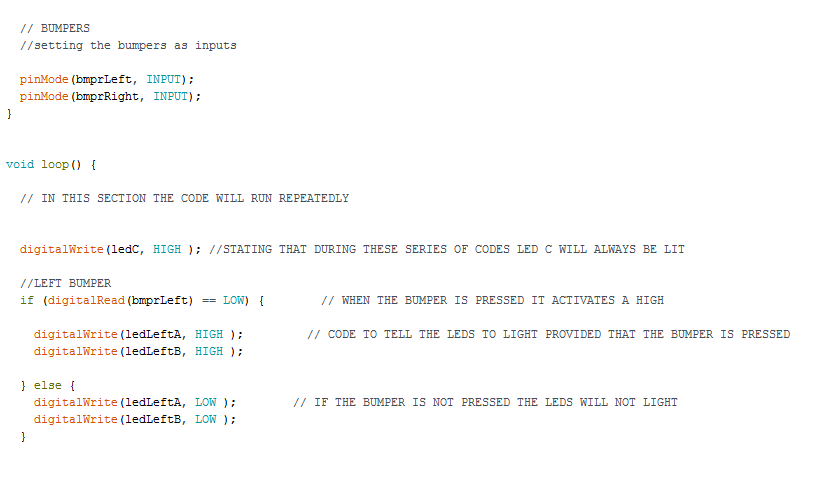
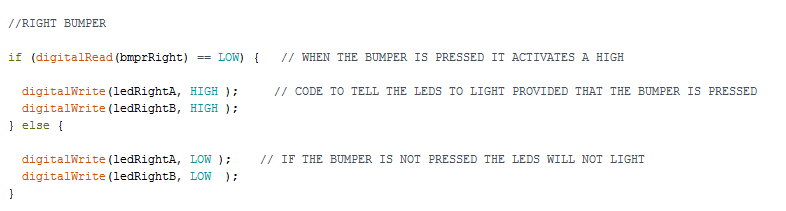
The bumpers on the robotic car are designed to detect the detection/s of object/s. On the Arduino, the bumper are set as inputs and have two states, either a low or high. The bumpers are controlled by the Arduino by the code inputted into the Arduino.

Method:

* First open the application Arduino
* In order to make the bumper code you need to open up a copy of the pervious code you made for setting the Leds and modified this code and this code will be saved as “bumperTest\_XY” (XY being your initials)
* In the first section, you need to assign the bumpers as two ’const int' (constant integers) statements to assign names to the two Bumper Switch pins. it is recommended to formulate a use short name that means something, for instance shorten the word ‘bumper’ to ‘bmpr’
* Then in setup, declare both Bumper pins as inputs.
* Afterwards for the loop section, you will need to the following:
* You'll need to use a digital Read instruction to allow the Arduino to check whether a switch is open or closed.
* In the loop () section you'll need to use and IF…else statement to turn on LEDs if the bumper switch is open or closed.

The code should end up looking something like this:





1.3 Making the robotic car drive

Objective: this code is designed to test the robotic car going forwards and backwards

When the robotic car is going forwards and backwards the Led should flash on and off repeatedly

Method:

For the robotic car to rotate a high must be sent to both motors on the Arduino

The the pin labelled 1Y must be high and the pin labelled 2Y must be low.  To do this the Arduino needs to output a high on D7 & a low on D8 and the pin labelled 3Y must be high and the pin labelled 4Y must be low.  To do this the Arduino needs to output a high on D9 & a low on D10.

This will enable the robotic car to go forwards

the pin labelled 1Y must be low and the pin labelled 2Y must be high.  To do this the Arduino needs to output a low on D7 & a high on D8 and the pin labelled 3Y must be low and the pin labelled 4Y must be high.  To do this the Arduino needs to output a low on D9 & a high on D10.

This will enable the robotic car to go backwards

1.4 Making the robotic car turn

Objective: to make the robotic car turn both clockwise and anti-clockwise

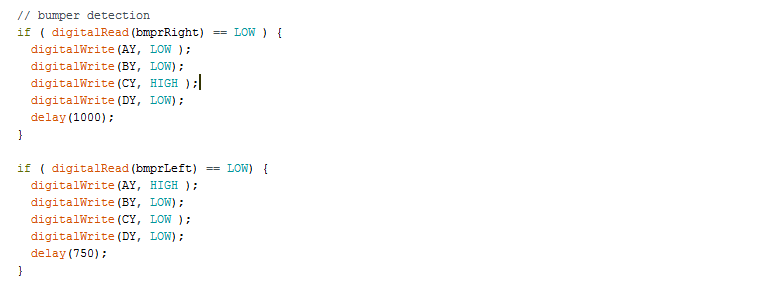
In order to make this code you must modify your pervious bumper code as the bumpers are fundamental to this code

Method:

For the robotic car to rotate a high must be sent to both motors on the Arduino:

* + - For the motor to rotate clockwise, the pin labelled 3Y must be high and the pin labelled 4Y must be low.  To do this the Arduino needs to output a high on D9 & a low on D10.
    - For the motor to rotate anti-clockwise, the pin labelled 3Y must be low and the pin labelled 4Y must be high.  To do this the Arduino needs to output a low on D9 & a high on D10.

The code should end up looking something like this:



When the robotic car is rotating all the leds should flash on and off repeatedly

1.5 controlling the speed

Objective: this code is designed to control the speed of the robotic car

The pmv (pulse wave modulation) ranges between o t0 255; 0 - 50 being the slowest speed and 255 being the fastest seed the car go at.

Pulse wave modulation allows variance in car speed

Method:

Aline of code like this needs to be inputted into pervious code Drive,

analogWrite(MotorLSpd, 100);

where MotorLSpd is replaced by whatever name you have assigned to the L293 enable pin.

1.6 The basic drill for the robotic car

Objective: this code is designed to test the culmination of all the codes written for the Robotic car

* Go forward slowly for a few seconds
* Go forward at a medium speed for a few seconds
* Go forward quickly

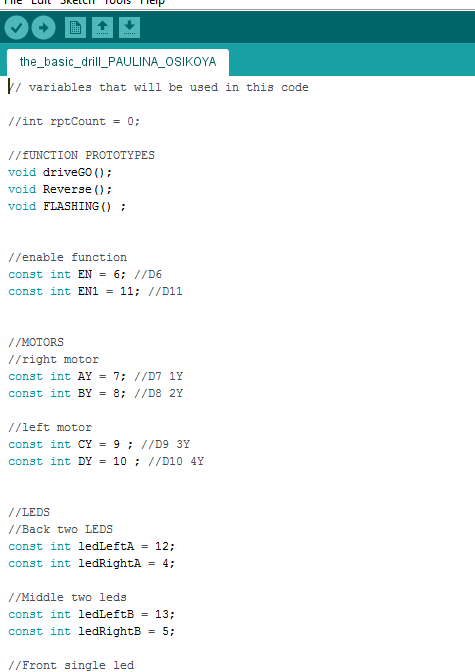
If, at any time, the left bumper is pressed Robotic Car should reverse slowly, turn approx. 45 degrees right and continue forward

If, at any time, the right bumper is pressed Robotic Car should reverse slowly, turn approx. 45 degrees left and continue forward

In this code, it is important to not neglect the leds as they will act as indicators for the robotic car’s motion

In this code functions have been added to the code and basically a function is a piece of code separate from the main code declared as a void function; that when called upon in the main code (void loop) it carries out its specific function

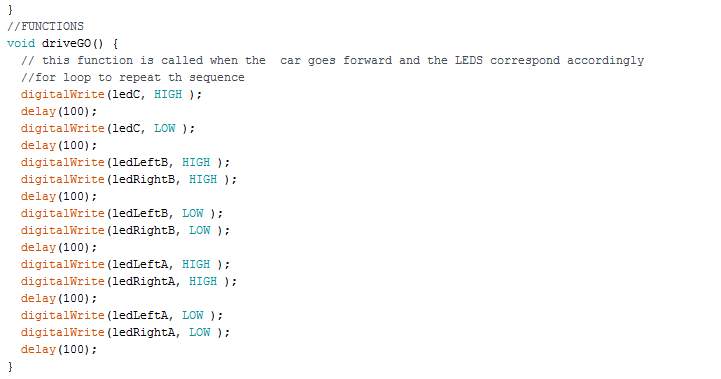
The code should end up looking something like this:

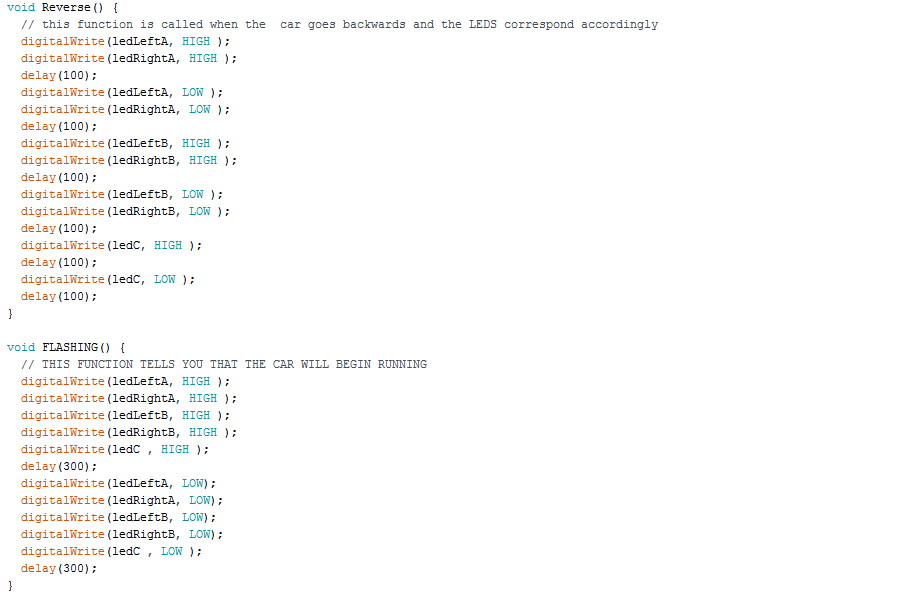












**Setting up the Robotic car**

Objective: to set up the robotic car, attaching the strip board and also attaching the appropriate wires to the Arduino and l293D chip and the external power sources or 6x 1.5v batteries to power up the robotic car.

Once you have set up your robotic car, you can now proceed to testing out your individual codes that you formulated in the section programming the Arduino in this manual.

Here you will be able to tweak your Arduino codes to functions properly, if they are not already fully functioning

For the final part of the physical construction, the motors, the Arduino and the stripboard components need to be connected

Here is the method for setting up the Robotic car:

* *the wires from the motors and switches are probably already connected to the white connector block on the robotic car*
* *Secure jumper wires into the 'other' side of the white connector block, if they are not already there with a screwdriver from your tool kit.  The other end of these wires will be used to connect to the connectors on the stripboard.* ( refer to the stripboard layout the 10 way connector section at the top half left side of the page)
* Then insert the other ends of those 8 jumper wires into the appropriate sockets in 10-way connector. This will connect the stripboard to the robotic car

\*If you get stuck Refer to the layout diagram to help you identify the sections In setting up the robotic car \*

* Next use 2 jumpers wires to connect the 0v socket of the 10-way connector to the Arduino GND pins (any ground pin can be utilized) and the Vin Arduino socket of the 10-way connector to the Arduino's Vin pin.

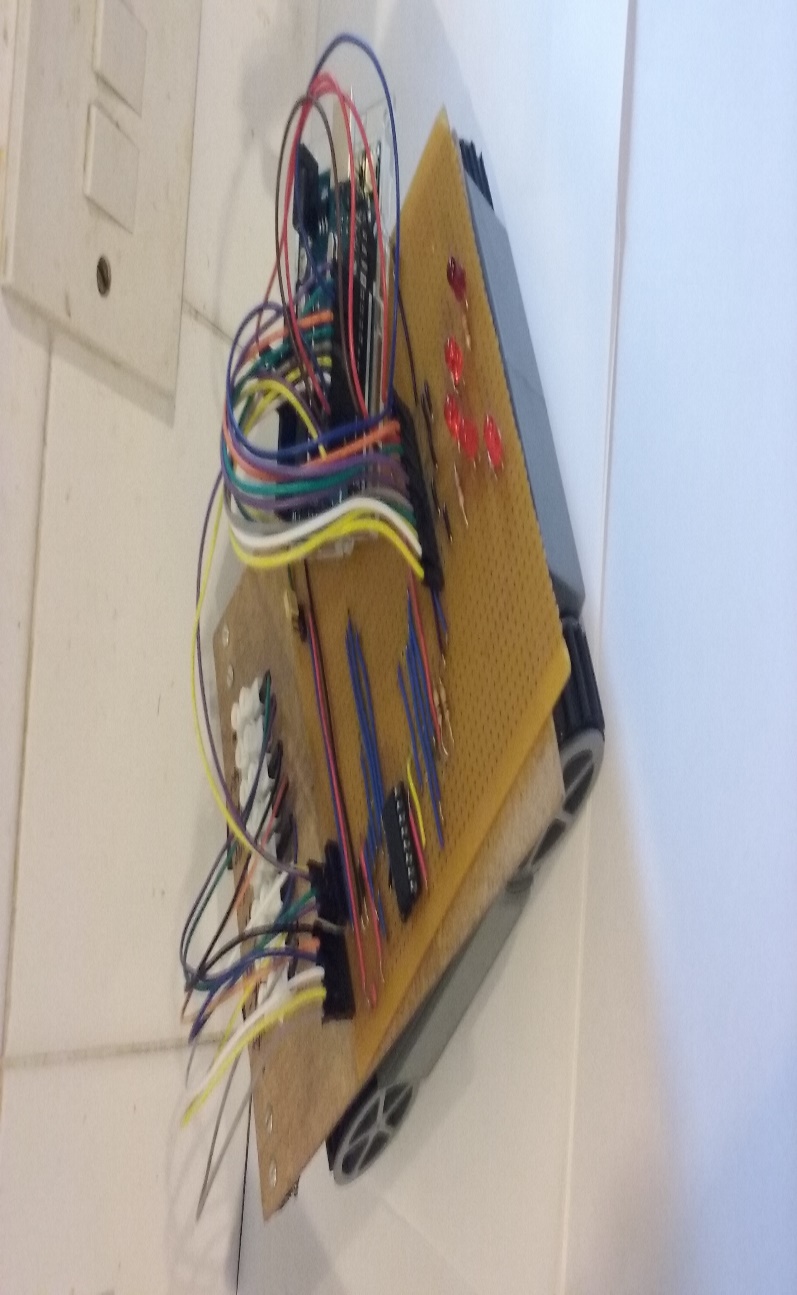
\*Once again, if you get stuck, refer to the layout diagram to help you identify the sections in setting up the robotic car \*

* Lastly, use more jumper wires to connect each of the pins on the 14-way connector to the correct Arduino digital pin and the 5v pin on the 14-way connector to the Arduino 5v pin

Tip:

A quick tip would be to install the batteries last into the robotic car as the car can still be powered by the external power sources and the Arduino, provided you connect the Arduino up to power via the usb cable. Placing the batteries in the batteries while still testing your code on your robotic car would be futile as it would only drain the battery. It is only when you are certain that your car code and connections are fully functioning that you can then insert the batteries, declaring the robotic car independent of external power sources.

*Another tip would be to follow the colour coding co-ordination for connecting the motor, bumper switch and battery wires to the firs side of the white connector block as it will make it easier to identify the parts.*



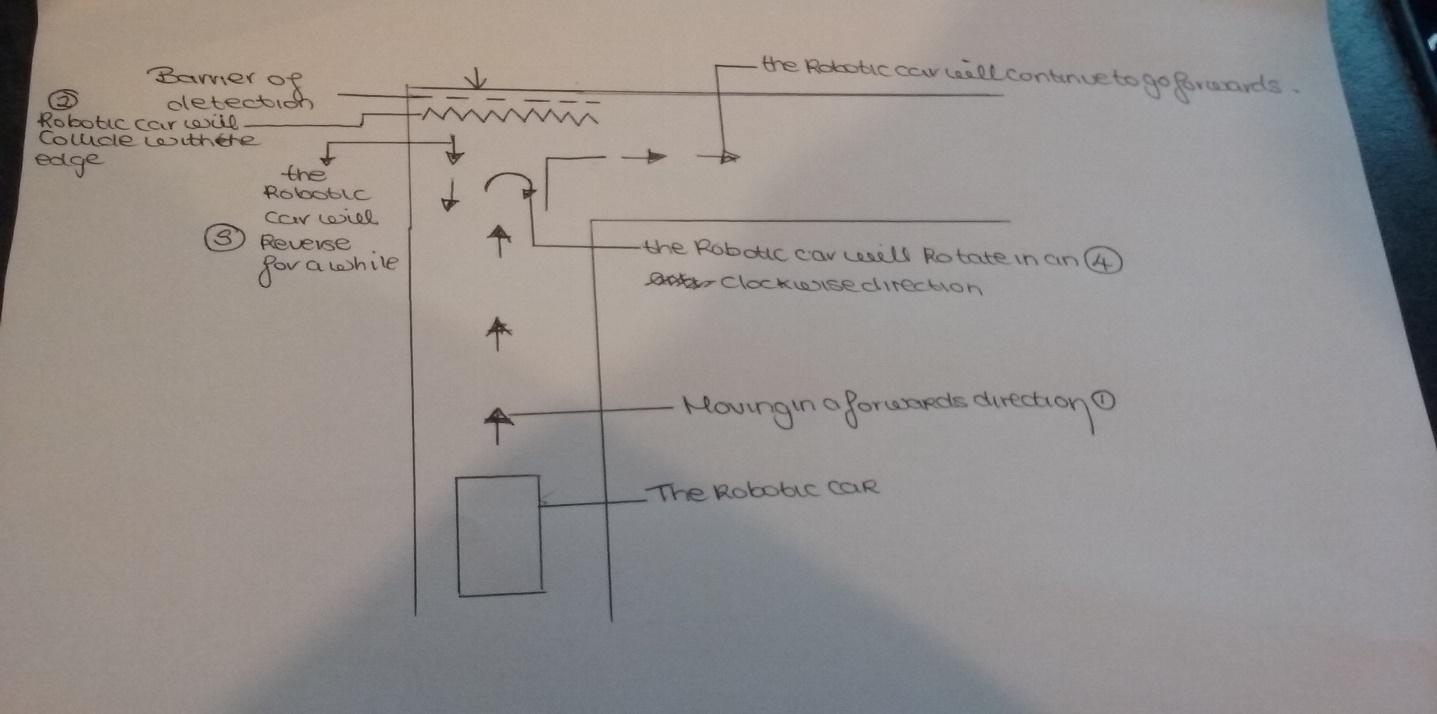
Implementation of the robotic car

Objective: in this section, the robotic car will carry out the basic drill of going forwards, detection of an object by the bumpers and reversing and in turn turning at a 45 degrees angle and then going forwards again.

To observe the implementation of the robotic, connect the first two pins on the 3-pin strip connector; this will utilize the batteries to power up the motors which will work in conjunction with the Arduino board and the l293D chip to carry the specific code constructed on Arduino

Once the motor begins, place the robotic car on a flat surface to carry out the basic drill. While the robotic car is carrying out the basic drill the leds should corresponded accordingly

This is how the path the robotic car should take in regards to the basic drill



Tip:

If the robotic car doesn’t carry out the basic drill it is important to check the following:

* The batteries may not be inserted. If this is the case please refer back to the section setting up the robotic car in the manual

* New batteries may need to be inserted.
* The 10-way pin connector is faulty and/or not solder correctly. If this is the case please refer back to the section soldering in the manual
* The code on the Arduino board is incorrect. If this is the case please refer back to the section programming the Arduino in the manual
* There is missing a connections/s If this is the case please refer back to the section setting up the robotic car in the manual