

# **ELEC1601 ASSIGNMENT 1 - Initial Report**

## **Project Information**

### **1. Nature of the problem**

The core project go through the whole semester will be assemble a PIC Buggy self-Assembly Kit (CHI007) robot and using basic assembly language write a program to navigate it go through a complex maze as well as return to the Start after it complete a run through the maze, the robot also required to be programmed to play a tune which should be able to do this at any point in running the maze or after completing a run through the maze. The last and most important is that the robot must finish its journey within maximum 7 minutes.

### **2. Available Equipment**

Each class will be divided into groups about 5-6 members and each group will be given the same Buggy Robot Kits includes:

- Pack of PIC Buggy parts;
- Download cable;
- 4 AA batteries;
- Pliers;
- Wire cutter;
- Cross-head screwdriver;
- Buggy manual

For flexibility, extra feature or equipment (no more than \$10) can also be added to the robot.

### **3. Problem analysis**

To finish the task, students are required by using any of the available tools mentioned above to solve the following problem:

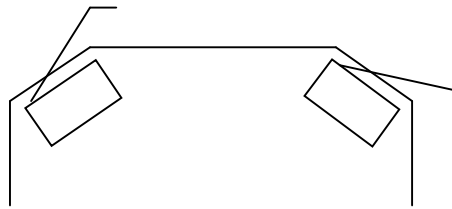
Assemble the robot physically, pliers and screwdriver is needed to integrate the PIC board with other external components as well as the soldering station will be used to link each component with the PIC board as there are about 22 soldering point need to complete.

Certain Logarithm need to be found and implemented into the program to achieve the highest efficiency, which will later be load into the robot via download cable and guild the robot go through the maze in limited time.

The music need to be played on the robot have to be converted into the machine readable code first and then decide how and when should it be executed.

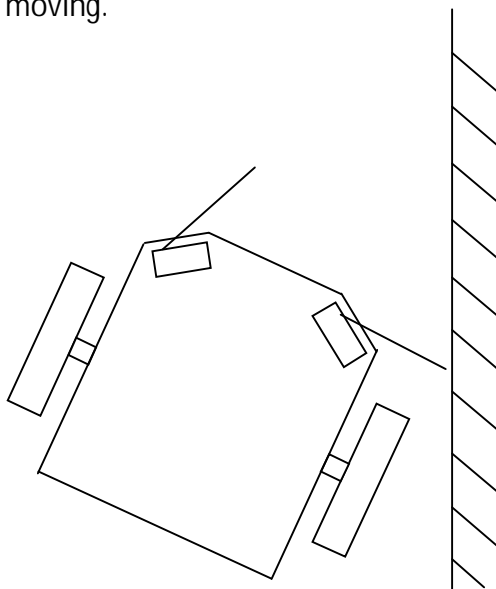
## Initial design

By observation, while the Robot is moving it will slightly move to its right hand side by itself, thus instead of adjust the rotating speed of its two wheels to moving fairly straight we can use this feature to go through the maze by design our robot keep moving along with the right hand side wall. To achieve this, our group firstly has to assemble the right hand side switch in a proper way which shows as the diagram below so its arm can detect there is a wall on the right hand side and the robot should be essentially programmed to automatically adjust itself movement in case it is getting too close or too far from the wall.

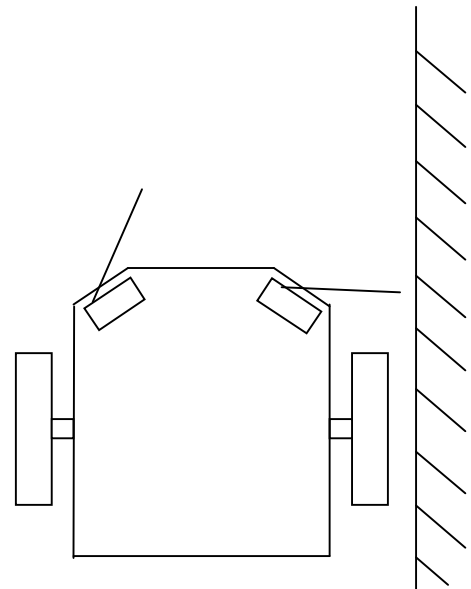


### Design 1: following the wall

According to our design, when the robot getting too close to the right hand wall, its right side arm will hit the wall and cause the switch on the right side of the PIC board been active, then based on the program that have been loaded to the robot previously, according to the certain condition the robot should perform the action such as stop and turn left slightly then keeping moving.



**Hit the right side wall**

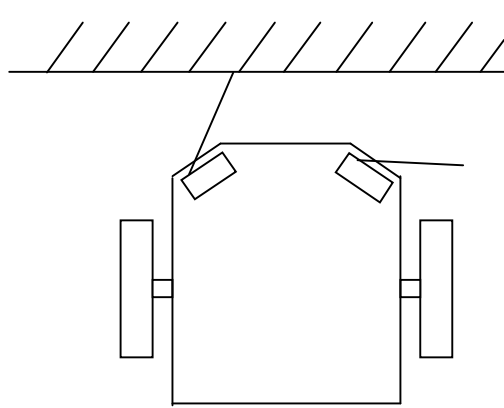


**Move along the right side wall**

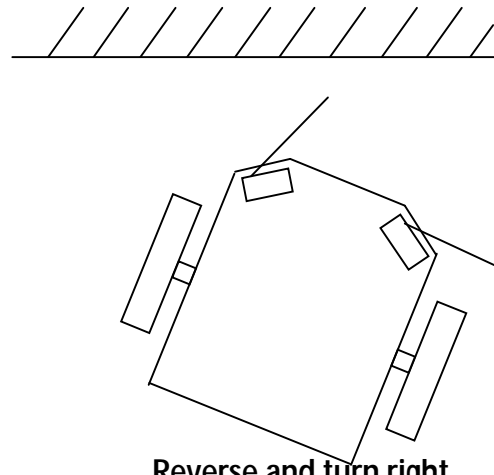
### Design 2: Going through a corner

When the left side switch is active, mean the robot is just hit a wall in front of it, in other words the robot is now encountered with an obstacle and have to find a new way to go through. In my design, as the robot hit the front wall it

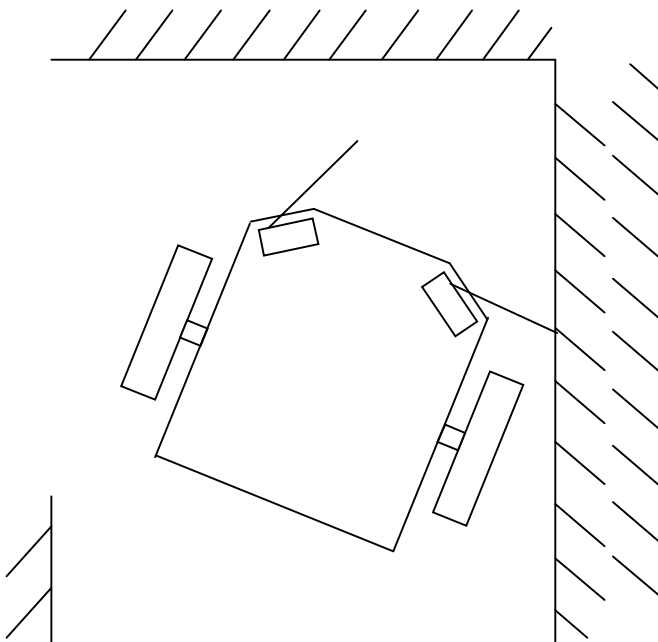
should firstly reverse and make a right turn, then the direction of the robot goes further will dependent on the result of its right turning.



**Hit the front wall**



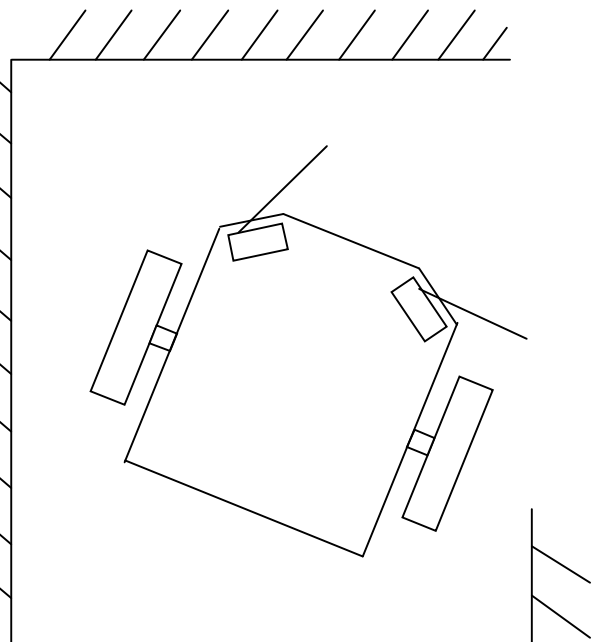
**Reverse and turn right**



**Condition One**

**Right side blocked**

**Right switch on**



**Condition Two**

**Right side clear**

**Right switch off**

Since the robot suppose to be moving on the right side of its rout next to the wall, therefore if there is still a wall on the right side of the robot, the right side switch on the robot will be active while it making a right turn and that signal will tell the robot there is a wall on the front and right side and the processor should give the correct instruction according to the program it have been set and finally turn to its left then keep moving.

If there is not wall on the right side of the robot, in that case the right side switch will not be turned on after the right turn. According to the program the robot should know that its right side is clear therefore turn future to the right and keep moving until hit another wall.

This is the core idea of my design and should provide the essential function to allow the robot go through the maze theoretically.

## Project plan & Task breakdown

The PIC Buggy would be physically assembled from each of its components in week 4's laboratory, therefore the following tasks will be focused on writing a proper program and load into the robot, so it can be executed while the robot is moving and playing a music while guiding it go through the maze.

Basically, our group can be divided into 3 parts each consist two members, one part focused on how to let the robot adjust itself keep it follow the right side wall while it is, one work on how to implement the idea about finding new way when it encounter an obstacle and the last part will take charge of making the robot play a tune.

These three processes generally can be processed individually as well as some codes still can be sheared and used for their different purposes, its detail will be discussed in further and presented in final design report on week7.

The final program will be the combination of the codes obtain from these three processes and planed to be finished and loaded into the robot at week 10, therefore certain test and change may be done before the final demonstration session come up.

	Part one	Part two	Part three
Week 5	Reading the assembly instructions and the example codes		
Week 6	Analysis and discuss the problems		
Week 7	"stop" action coding	"left turn" action coding	Convert the music into code
Week 8	"right turn" action coding	Import "stop", "right turn" action code	Decide the condition for playing
Week 9	If statement coding	Nest if statement coding	Import if statement
Week 10	Integrate the codes, load into robot and testing		
Week 11	Demonstration		

## Potential difficulties

1. Students have no experience on assembling or programming a robot using unfamiliar tools and language.
2. According to our initial design on the robot, since it is aimed to move along the right side wall while it is moving through the maze, it can be difficult to keep tracing the wall during its journey.
3. As the robot reaches a corner, it has to find the correct way to continue its journey; this feature is expected to be worked on the idea that mentioned in the initial design and the actual implementation could be a difficulty during the assembly.
4. From the final report been carrying out to the demonstration, there is only 3 weeks for the robot to be assembled. If any part of project after schemes, it can cause a major delay on the final product and later work is not acceptable.

## Possible Strategies

1. Students are expected to read the assembly instructions in the PIC Buggy manuals and the example code from the unit of study web page carefully to get the basic idea of assembly, any confusion or doubting should be issued and discussed on group meeting or ask tutor if necessary.
2. If statement could be used during programming, if the right switch is on, the following code should be executed: stop, right motor running; to correct the heading of the robot parallel to the right wall thus this statement should be executed at any time while the robot hit the right side wall and therefore keep it moving along the wall.
3. In this case, a nested if statement is required as the first if statement determines whether the left switch is on or not and if that is true then an action will be taken as well as the second if statement which determines the final direction the robot goes dependent on the outcome of that action.
4. Since time may become a major issue in later of the project assembly, therefore each team member needs to work together and take part in the project from the first day till the end, everyone should strictly follow the task breakdown and project plan, no time should be wasted.