# Post-Lab 1 Report Alan Palayil

ECE 100-00X Teammates:  Mathew, Rezwan

Prof. Oruklu Lab Date: 8/30/19

TA: Rafael Alejandro Perez Due Date: 9/06/19

## Problem Statement

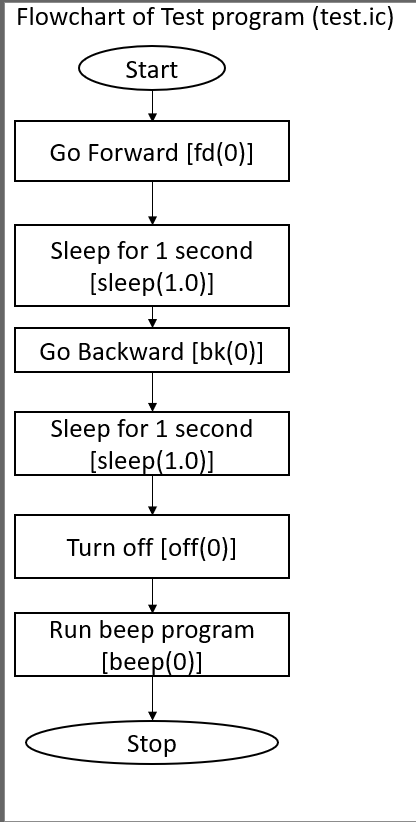
            To read the introductory chapter of basic programming of Interactive C and building a simple handy-bot robot which can avoid an obstacle while completing the maze. To construct Handybug using Handy-board and program the Handybug to avoid an obstacle.

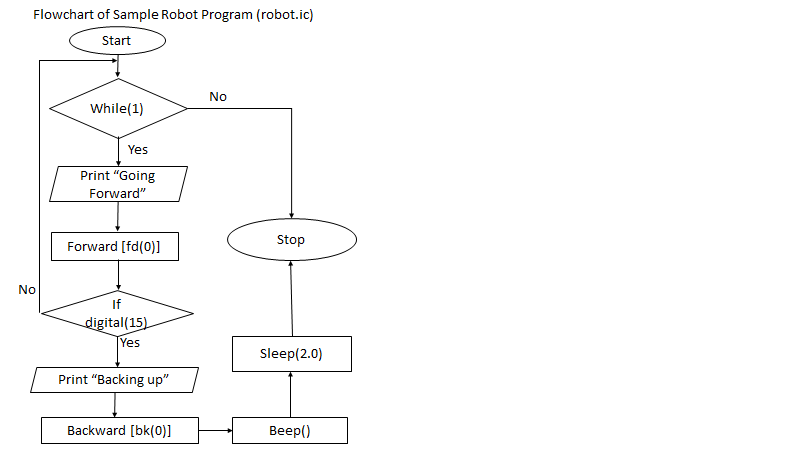
**Investigation/Research**

            The chapter introduces the Interactive C programming. There are different program functions explained. The complete handy-bot board is explained. The programs such as to test the board, motors and the sensors provided are working will be tested. The building of the handy-bot with basic motor functions and we will try to make the handy-bot avoid an obstacle. To avoid an obstacle the basic idea is to place the sensors in front of the handy-bot and when the bot crashes on an obstacle, the bot such move backward then turn in order to avoid the obstacle and move forward. The basic logic is when the bug hits an obstacle in the left, it should turn right and when the bug hits an obstacle on the right, it should turn left. The build started with following the basic instructions to build the layout of the Handy-bug. After completing the build, trial programs were run.

**Alternative Solutions**

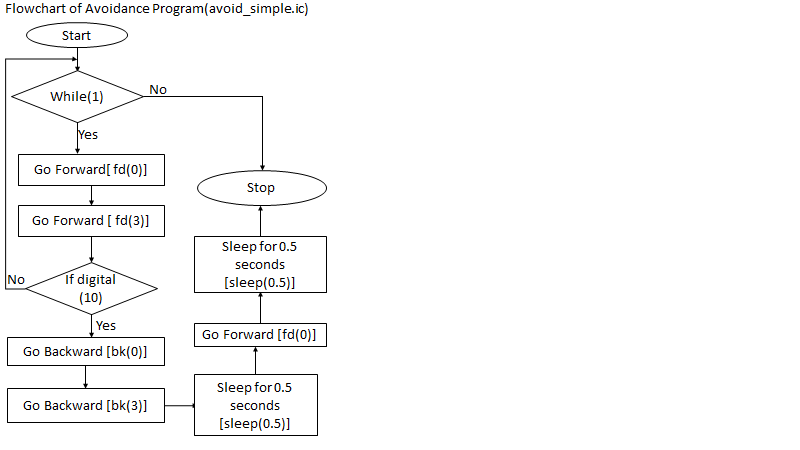
       The flowcharts are provided.





The handybug can move in one direction again and again to change its direction even if either one of the sensors are activated.

**Optimum Solution**

The most optimum solution for the handy-bot to avoid an obstacle. The solution involves the handy-bot to move forward as long as there is no obstacle in front of it. When an obstacle is in front of the handy-bot and the analog sensor gets activated and the handy-bot stops, moves backwards then later will change its direction slowly and then moves forward and in turn should prevent the obstacle.

 For the handybug to solve the maze quicker the bug requires to commute in the opposite direction to where the sensor got activated.

**Construction/Implementation**

            Before construction and implementation can take place, the team decided to make the Handy-bug as per the instructions. Building the structure from scratch helped in understanding the purpose of each component. Connecting the cables between the handy-board and motor. The motor 1 cable was connected to port 0 and motor 2 was connected to port 2. The bug moves using the two motors and helps it to move forward and backwards. The touch sensors are placed in the front of the bug. Since touch sensors are digital sensors, they are connector to the digital ports 7 and 8. Port 7 sensor is placed in the left and Port 8 sensor is placed in the right. Using just one motor the bot can turn either to the left or right. So, if the bug hits an obstacle then the bug will go backwards for 5 seconds. When the left sensor gets activated the right motor turns off and when the right sensor gets activated the left motor turns off.

**Analysis & Testing**

           While observing our design in action, we test whether the motors were working. If the sensors were responding to touch or hit an object. The basic testing program was loaded to check whether all the components were in working conditions. After downloading the final program into the board. When the bug’s program ran, we noticed that one motor was faster than another and that the bug was moving inclined towards the right. So, we had to create a motor function which decreased the rotational speed both forward and backward. After several trials, the bug’s movement became straight and for turning movements the time of delay was changed to 0.75 seconds.

**Final Evaluation**

            The final program and the Handy-bug should be able to complete the maze within the requested time.

**References**

1.  Martin, Fred G. 2001. *Robotic Explorations: A Hands-On Introduction to Engineering*. New Jersey: Prentice Hall.

2.  Oruklu, Erdal. 2015. *ECE 100* *Lecture Notes*. Chicago: Illinois Institute of Technology, Electrical and Computer Engineering Department.

**Attachments**

List your attachments here (signed lab notes, avoid\_simple.ic, etc.) and staple them to your report.

/\* Sample robot program \*/

void Forward ()

{

    motor (0,45);}

void Backward ()

{

    motor (0, -45);}

void main () {

    while (1) {

        printf ("Going forward\n");

        Forward ();

        fd (2);

        if (digital (9)) {

            printf ("Backing up\n");

            Backward ();

            bk (2);

            beep ();

            sleep (0.5);

            /\*Turn\*/

            fd (2);

            sleep (0.75);

        }

        if (digital (7)) {

            printf ("Backing up\n");

            Backward ();

            bk (2);

            beep (); sleep (0.5);

            /\*Turn\*/

            Forward ();

            sleep (0.75);

        }

    }}