

# TurtleSim Bot

Dipti Ranjan Sahu	I40070038
Abhay Vikram	I40040113
Rohan Vora	I4D170003
Surya Teja	I40070048
TEAM ID: 394	

# Problem Statement

The problem at hand is to make a hardware implementation of a small number of commands of the function TurtleSim in simplecpp library which would enable a user to draw basic shapes on paper.

- Our most basic goal, was to design a program, which would read the code written by the user, and give instructions to the bot accordingly. The bot, would then trace out the path of the shape that would have been constructed virtually, had the program been compiled on a computer. A pen would be attached to the bot and controlled by a servo motor, allowing the pen to draw a shape as required.

# DIVISION OF PROJECT WORK

1. Dipti wrote the code for the communication of commands from the laptop to the bot. He also wrote the code taking input from the .txt file.
2. Rohan wrote the code for identifying keywords like TurtleSim, forward, repeat, left, right etc. in the user entered program and issuing commands to the bot accordingly.
3. Abhay dealt with the installation of the pen and the servo motor. They had to be installed with precision so that upon the turning of the bot, a discontinuous line would not be formed. This would reduce the amount of coding that had to be done.
4. Surya, wrote the code for accepting an integer value as an argument, and precisely getting the bot to move by that amount in mm (or in the case of turning, getting the bot to turn by the given amount in degrees)



# Project Screenshots



# Challenge

- The following challenges were encountered by us during this project.
- ‘Translating’ commands from simplecpp to Embedded C: We converted the code for the program into a .txt file and read it using input functions in simplecpp. We searched for particular strings like “forward”, “left”, “right”, etc. and upon finding them, read their arguments as strings, converted them into integers, and passed these numbers as arguments into our drawing functions.
- Location of the servo motor: Initially, the servo motor was placed on the sides. However, this led to the writing of unnecessary amounts of code to prevent a discontinuous/deformed shape being formed. This was resolved by placing the servo motor such that the tip of the pen was as close to the axle as possible.
- Xbee Communication: Originally, we were thinking to send the entire code via Xbee to the bot, which would then process it accordingly and give the required output. Afterwards, we realised that we could only send characters one-by-one and therefore we dropped this idea and evaluated the code in simplecpp, and sent arguments character by character.

# Future Work

- Some modifications possible for our bot are
  - Further features of C++ library like data-types are not read in our code. We will try to improve it and add identification of data-types like int, float, etc. in our code.
  - Functions like cout in C++ could be displayed on the bot's LCD screen.
  - Attaching another servo motor to the bot, would enable it to erase any patterns as well (given that they are not permanent). A brush/duster could be attached to the bot through the servo.
- 
- Our project has design, artistic and teaching applications. Some of them are:
  - It can be extended as a drawing bot, where you design the drawing/sketch on your computer and the bot makes it for you.
  - Just like TurtleSim was a learning tool for beginners to C++. This bot can also be a fun and educative way to begin learning the language.





THANK YOU !