System Requirements Specification for: Embedded Systems Project: A-star based grid traversal using Firebird V

Bhuvan Teja 09005049
Varun Suprashanth 09005063
Salman Khan Pathan 09005064
Asok R 09005072

February 17, 2012

1 Introduction

1.1 Overall Description

The project is aimed at making the firebot detect the shortest path in a grid with unknown obstacles using the A-star search algorithm. For simplicity's sake this project is to be carried out assuming the grid is a surface with whitelines for path and black areas as obstacles.

1.2 References

References to other documents/sources

1. A.I. course by Pushpak Bhattacharya
2. Wikipedia
3. Pololu Robots and Electronics(Website)
4. Robot Navigation by Ramakrishna Bairi guided by Prof.Pushpak Bhattacharya

1.3 Overview of Developer's Responsibilities

Developer is responsible for development,installation and loading of all required programmes or data. He is responsible for checking that the robot satisfies all the required conditions. Developer must check for various possibilities to be covered by the bot and also should account for all the jobs to be performed by the bot. Developer also must ensure that the algorithm given is followed correctly and bot must work fine for any maze.

2 General Description

2.1 Product Perspective

The firebot will interact with the grid using the infrared sensors and the data thus produced is used in the A-star search. The maze can be of different types. The way bot senses maze depends on the type of maze. If the maze is made of distinct physical borders, infrared sensors are used for sensing the maze. If the maze is a black line, whiteline sensors are used for sensing the maze. Also, to scout the algorithm, if the maze is black line maze, we can actually use the famous 3pi robot algorithm for scouting the maze and then A-star for finding the path.

2.2 Product Functions Overview

Since the successful completion of the project would imply minimal human-bot interaction mentioned below is an abstract list of some automated bot functions in various situations.

1. Scouting
2. Grid Mapping
3. Shortest Path Detection
4. Shortest Path Traversal

2.3 User Characteristics

The end-product is useful for a range of users.The bot can replace human presence where there is a chance of danger eg:bomb difusal needs the bot to traverse through rubble or unknown conditions.Nuclear plants or extra terrestrial excursions can and already make use of something similar to this.Even when the layout is known it can be fed to the robot which will traverse on the shortest path available-this situation might be more practically applicable.

2.4 General Constraints

The time taken to complete detecting a shortest path might be a issue. And where the demo is concerned there might be issues with the bot movement. Also the surface on which the maze is made also effects the motion of the bot. There also need to be many more constraints like the speed of the bot, which should be low while scouting and high when moving in the shortest path, increasing effeciency and also increasing precision and saving power. Also, depending on the bot, power used by the bot should be minimal, so as to make the bot very efficient.

3 Function Description

As described before in the overview the functions are rather steps in which the bot progresses towards it's goal.This section actually gives a bit more description on what the functions of the bot are and also what are essential in the operation of the bot. Descriptions for each of the steps is given below

3.1 Scouting:

Traverse the entire grid to obtain data for mapping. This is done by either A-star algo or the 3 pi algo. Any algo, just sets some constraints on the bot as to which case to prefer over which other. This does all the checking and saves the data as which nodes are present on the shortest path. This is the very first part of our maze solving, and also the most important one, since not missing any nodes during scouting is a valid concern to be taken care of.

3.2 Grid Mapping:

Map the data obtained i.e change it into a form such that the processor can use it. The data obtained is w.r.t what operation it should perform in what case so that Algo is followed correctly.

3.3 Shortest Path Detection:

Uses the mapped grid to obtain the shortest path through A-star algorithm. This is the trickiest part. In this part, use of data structures comes in processing the data to find the shortest path. We will have to write c code of finding the shortest path, transform it into esterel code and load it on the bot. There, it is very important that the code must be efficient because this is the part of optimisation in the bot.

3.4 Shortest Path Traversal:

Use the path given to move from end state to start state. In this part, several things are to be taken care of. For example, if it is a straight line for long, but must accelerate and go at a higher speed. If there is a turn, it must go at a lower speed. Speed must be managed precisely because bot may go off the path, and may lead to inefficiency and non-traversal of the shortest path.

4 External Interface Requirements

4.1 User Interfaces

No user interface as such.User interaction with the bot is direct and kept to the minimum. Once the bot finishes scouting and reaches the end point, user has to put it in the start position and then should start the bot again.

4.2 Hardware Interfaces

Firebird Robot
Atmega 2560
Infrared Sensors
White line Sensors

4.3 Software Interfaces

Atmel AVR Studios

5 Performance Requirements

Performance can be measured only after completion.Expected to be in the range of 1-5 minutes depending on the grid complexity. Largely depends on type of maze and type of surface on which the maze is made.

6 Design Constraints

6.1 Standards Compliance

Not applicable.

6.2 Hardware Requirements

Requirements won't exceed those available in lab.A grid is to be constructed. We require a firebird bot.