

Arduino based obstacle avoidance robot

Report Name	Outline Project Specification
Author (User Id)	Daniel Atkinson (daa9)
Supervisor	Dave Barnes (dpb)

Module	CS39440
Degree Scheme	G401
Scheme Name	BSC Computer Science

Date	October 22, 2012
Revision	1.0
Status	Release

1 Project description

When making an autonomous robot, regardless of its primary function, it will need to be able to avoid damaging itself on its environment, regardless of how expensive the robot may be, or somehow restricting itself. This can be in the form of a reactive system or something more intelligent. To not get too close to an object in the environment or even in some cases not colliding with itself, such as in the case of an arm.

The aim of the project is to design and build a wheeled robot that can independantly (with no user input) move around within an environment and not get hindered by any obstacles that may be in its path.

This will be done by building a bump skirt out of sonar or infra red sensors to avoid objects in direct proximity. These will be placed all around the robot. The other method will be some more of these sensors mounting in such a way that they can move to look around the environment and attempt to plan the next direction to move and possibly a route.

2 Work to be tackled

2.1 Hardware

Some of the work to be done will be designing the hardware of the system. This would have to take into consideration things like battery life expected for a 'decent' runtime, overall weight so that it is not over encumbered, size to make it practical and speed of operation as to not take too long and become useless or a hinderance to its purpose.

The various pieces of hardware or even the various subsystems will need to be able to talk to each other reliably.

- Hardware Research
Research will have to be done into the different ways this can be achieved. Also a method to auto deploy updates to all the subsystems without any manual hardware reconfiguration each time will be useful. This may require some additional initial hardware setup and will also need researching.
- Testing
Due to the hardware component to this project planning detailed tests will be needed. Automatic unit testing will be a problem and due to the varied sensor readings obtained in a real world environment as opposed to that of a simulation, testing each system in a real environment would be appropriate.

2.2 Software

Software is the major component to this project where all of the sub-systems will be independant entities that do their own job and communicate with the other systems to form a complete system. These various systems may be running on seperate pieces of hardware and even written in different languages, this must also be taken into account.

Researching how to get these to communicate with each other is a key component to this project. Three main things need to be researched for this project in regard to software.

- Subsystem interaction
How to get the various hardware systems to talk to each others software systems.
- Sensor data
How to get the data required from the hardware sensors and make that information useful.

- Processing
Making good use of the data collected and using that to control the relevant components and perform their tasks.

3 Project deliverables

- Chassis
To have the chassis built
- Development system
Install a computer onboard the robot to control all the subsystems and to do all the main processing, build and deploy updates to the attached systems.
- Movement
To be able to move around in its environment under its own power.
- Detect Object
For the sensors to be able to detect if something is close to them.
- Avoid object
Combine the movement and object detection subsystems to move around and not bump into obstacles.
- Find a path
Use directional sensors to try and find a path the robot will fit through.

4 Initial bibliography

- <http://lwk.mjhosting.co.uk/?p=343>
title: Getting Started with Raspberry Pi GPIO and Python
date accessed: 19/10/2012
- <http://absences.sofianaudry.com/node/10>
title: I2C communication between two Arduino boards
date accessed: 20/10/2012