Abstract

Acknowledgement

Index

[Abstract 1](#_gjdgxs)

[Index 2](#_1fob9te)

[Chapter 1.](#_3znysh7) Introduction 4

[1.1](#_2et92p0) Introduction 4

[1.2](#_tyjcwt) Problem Definition 4

[1.3](#_3dy6vkm) Scope 4

[Chapter 2.](#_1t3h5sf) Literature Review 5

[Chapter 3.](#_4d34og8) Project Management Plan 6

[3.1](#_2s8eyo1) Feasibility Analysis 6

[3.2](#_17dp8vu) Lifecycle Model 6

[3.3](#_3rdcrjn) Project Cost and Time Estimation 6

[3.4](#_26in1rg) Resource Plan 6

[3.5](#_lnxbz9) Task and Responsibility Assignment Matrix 6

[3.6](#_35nkun2) Project Timeline Chart 6

[Chapter 4.](#_1ksv4uv) Project Analysis and Design 7

[4.1](#_44sinio) Software Architecture diagram 7

[4.2](#_2jxsxqh) Architectural style and justification 7

[4.3](#_z337ya) Software Requirements Specification Document 7

[4.4](#_3j2qqm3) Software Design Document 7

[Chapter 5.](#_1y810tw) Project Implementation 8

[5.1](#_4i7ojhp) Approach/System Architecture / Main Algorithm / Methodology 8

[5.2](#_2xcytpi) Programming Language used for Implementation 8

[5.3](#_1ci93xb) Tools used 8

[5.4](#_3whwml4) Deployment diagram 8

[Chapter 6.](#_2bn6wsx) Integration and Testing 9

[6.1](#_qsh70q) Testing Approach 9

[6.2](#_3as4poj) Testing Plan 9

[6.3](#_1pxezwc) Unit Test Cases 9

[6.4](#_49x2ik5) Integrated System Test Cases 9

[Chapter 7.](#_2p2csry) Conclusion and Future Work 10

[Acknowledgement 14](#_30j0zll)

[Index 15](#_147n2zr)

# Introduction

## Introduction

## Problem Definition

## Scope

# Literature Review

# Project Management Plan

## Feasibility Analysis

## Lifecycle Model

## Project Cost and Time Estimation

## Resource Plan

## Task and Responsibility Assignment Matrix

## Project Timeline Chart

# Project Analysis and Design

## Software Architecture diagram

## Architectural style and justification

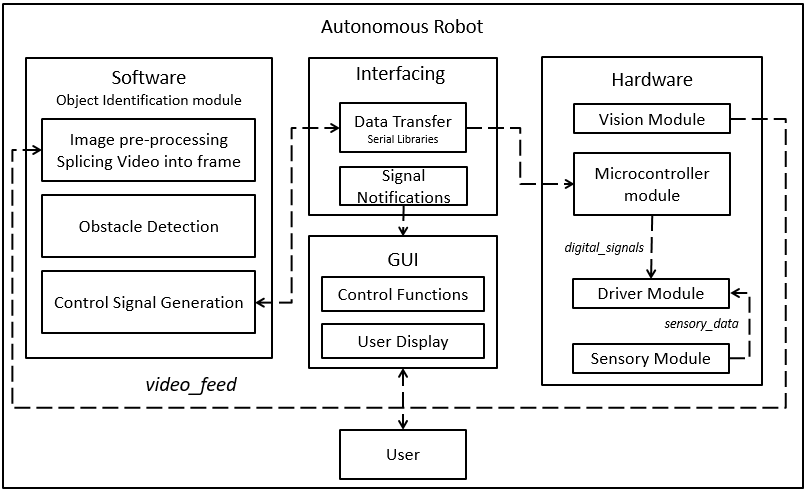
## Software Requirements Specification Document

## Software Design Document

# Project Implementation

## Approach/System Architecture / Main Algorithm / Methodology

The given diagram depicts the distribution of the various modules of the system. The flow of control and processing is shown via the interconnecting arrows between the modules which basically explains how the system processes the input image, identifies the object present in it and forwards the robotic operations via the controller to the hardware.

The system is broadly classified into two major working areas as the Hardware and Software. Different modules are then designed under these areas. An interfacing link is created between the two areas to allow communication and flow of data.



The GUI controller will display real time data to the user and is included as a module for direct human interaction limited to only monitoring of the system and not involving computing interaction.

### The software layer

### *Image pre-processing*

This module directly receives the video feed of the surroundings from the Vision module. This video feed is split into multiple frames at some intervals and forwarded to the obstacle detection module.

### *Obstacle detection*

This module analyses each of the frames split from the video feed and detects and identifies the obstacle using the classification function. This

classification function compares the obstacle with derived feature vectors

and identifies it on the closest match.

### *Control signal generation*

Depending on the type of obstacle identified, this module sends appropriate data(flags) to the Interfacing layer which converts them and sends them to the microcontroller.

### The Interfacing layer

## *Data Transfer*

This module receives control signals from the Control signal generation module and converts them into Serial data which can be understood by the microcontroller using Serial libraries. It communicates with the hardware via a Serial port.

### *Signal notifications*

Upon classification of the obstacle, the category is sent to the User Display via this module. This module notifies the user about the output of the obstacle identification via the GUI.

## The Hardware layer

## *Vision module*

This module receives the video feed from the web camera attached to the robot. The primary function of this module is to receive input about the surrounding area from the camera and forward it to the Image pre – processing module.

### *Microcontroller module*

This is the most important module in the system as it governs the working of the robot, decides its movement based on the control signals received and coordinates with all of the other hardware module using digital signals.

### *Sensory module*

This module receives input from the line sensors which are used to guide the robot on a pre-defined path and range sensors which are used to determine the distance between the robot and the surrounding obstacle. This sensory input is sent to the Driver module for navigation and control.

### *Driver module*

This module controls the motors of the robot and decides the motion, direction and speed. It receives input from the sensory module which notifies it about the path and the existence of surroundings obstacles.

### The GUI layer

### *Control functions*

This module defines several control functions which allow the user to control the bot’s movement in case human intervention is required. This module is necessary as it ensures manual control over the robot in case of emergencies such as battery failure, damage to the robot and error in processing.

### *User display*

This module displays the video feed of the web camera mounted on the robot.

## Programming Language used for Implementation

### MATLAB

The primary reasons MATLAB was chosen for implementation:

1. Strong mathematical and numerical support is provided for the implementation of advanced algorithms.
2. Developers can have total control of precision for data representation in MATLAB. An image can be casted into floating point values with double precision.
3. MATLAB allows developers to test algorithms immediately without recompilation. Algorithms can be tested without recompilation. Results are displayed instantaneously thus saving a lot of crucial execution time.
4. MATLAB contains many built in functions which provide a wide array of image processing functions and can be easily read and understood.Due to this feature, MATLAB is clearly distinguished from applications such as Photoshop.
5. Toolboxes provided by MATLAB such as the Computer Vision and Image Processing toolbox have been crucial in the execution of image classification and identification tasks.
6. Interfacing with hardware devices such as Arduino microcontrollers and other languages is also supported.

### Arduino

Primary reasons for choosing Arduino:

1. The main interfacing between hardware and software is provided through system integration. Arduino provides the Serial Library that allows code integration and data communication between MATLAB and Arduino IDE.
2. The Arduino library is a vast collection of established functions and forums which enable the user to easily create and debug new projects.
3. Connection with a wide array of co-processors and drivers (such as the L293D motor driver used in this project to control the motors) is supported to a great extent allowing the developers to control many hardware elements such as motors and sensors with relative ease.

## Tools used

## Deployment diagram

# Integration and Testing

## Testing Approach

## Testing Plan

## Unit Test Cases

## Integrated System Test Cases

# Conclusion and Future Work

References

Appendix

## Testing Approach

## Testing Plan

## Unit Test Cases

## Integrated System Test Cases