

Software Requirements Specification CS101 Projects 2015

SketchBot using Firebird V

Group CUSE

**Varunesh Goyal 140070006
Saurabh Pinjani 140070056
Saurabh Chavan 14D070036
Devesh Khilwani 14D070045**

Table of Contents

- [1. Introduction](#)
 - [1.1 Definitions, Acronyms and Abbreviations](#)
 - [1.2 References](#)
- [2. Overall Description](#)
- [3. Details](#)
 - [3.1 Functionality](#)
 - [3.2 Supportability](#)
 - [3.3 Design Constraints](#)
 - [3.4 Interfaces](#)
 - [3.4.1 User Interfaces](#)
 - [3.4.2 Hardware Interfaces](#)
 - [3.4.3 Software Interfaces](#)
 - [3.4.4 Communications Interfaces](#)
- [4. Quality Control](#)
 - [4.1 Test Data](#)
- [5. Risk Management](#)

1. Introduction

1.1. Definitions, Acronyms and Abbreviations

OpenCV-OpenCV (*Open Source Computer Vision*) is a library of programming functions mainly aimed at real-time computer vision.

ZigBee-ZigBee is used to create personal area networks.

ATmega2560-ATmega2560 is the micro-controller used in the bot and belongs to the family of AVR based micro-controllers.

AVR-AVR stands for **A**dvanced **V**irtual **R**ISC(Reduced instruction set computing). It represents a class of micro-controllers with a particular kind of design and architecture.

1.2. References

We mention here the list of the documents which find mention in our SRS. These have been (directly or indirectly) used as resources-

- 1.<http://kevinhughes.ca/tutorials/opencv-install-on-windows-with-codeblocks-and-mingw/>
- 2.<http://en.wikipedia.org/wiki/OpenCV>
- 3.<http://opencv.willowgarage.com/documentation/cpp>
- 4.www.atmel.com/webdoc/atmelstudio/

2. Overall Description

Product Perspective Automated Drawing robot

Product Functions The project is aimed at using image processing techniques to reproduce simple images comprising of lines on paper.

User Characteristics Someone requiring large scale sketches but lacks the facility to print them.

Constraints Input needs to be not too complex

Assumptions and Dependencies There will be sufficient friction between the drawing surface and the wheels of the bot to avoid slipping, and that the surface is flat without any obstacles.

Requirements Subsets Firebird V, Servo motor, Zigbee module

3. Details

3.1. Functionality

- An image will be taken as input using a simple User Interface that will help the user to browse the file and provide it as input to the program.
- The image will be processed to prevent broken lines and remove noise.
- A final image that has a continuous non intersecting path will be extracted from the input image.
- This image will be transformed into a set of instructions that will enable the robot to move so as to reproduce the path in the physical world .
- The instructions will be conveyed to the robot using wireless communication
- The robot will have a marker attached to it that will enable it to draw the path on the chart paper.
- If time permits, we will try to draw curved trajectories also.

3.2. Supportability

- The code will be readable, reusable and portable (The PC side algorithm).
- It shall be properly indented.
- The project will follow the Firebird specific guidelines.

3.3. Design Constraints

- The lines of the image should be contiguous and non-intersecting.
- The lines of the image should be straight lines.
- The marker used for drawing should have minimal friction.
- The wheels of the bot and the surface must have sufficient friction so that there is no slipping between the wheels and the surface.

3.4. Interfaces

3.4.1. User Interfaces

1. Ask the user to specify the image file which has to be drawn.
2. View the image file.
3. View the final image (after image processing) that will be drawn by the bot.
4. View the sequence of movements.
5. Signal the bot to trace the generated sequence of movements.

3.4.2. Hardware Interfaces

1. ZigBee module
2. Servo motor/Linear actuator

3.4.3. Software Interfaces

1. OpenCV:
Documentation for Open Computer Vision for C++:
<http://opencv.willowgarage.com/documentation/cpp>
2. ZigBee driver

3.4.4. Communications Interfaces

The ZigBee interface will serve as the communication interface.

4. Quality Control

4.1. Test Data

Our bot will first and foremost be able to draw simple geometric figures like regular polygons and their derivatives (curves that can be made using straight lines).

5. Risk Management

- Accuracy of final drawing has a possibility of error. It may happen that the bot gets deflected by small but significant amounts eventually resulting in distortion of image. In that case we shall reduce the scale of output drawing, even if this doesn't work we plan to make the input more simpler.
- Our project involves some image processing and transfer of information over a ZigBee module which we are yet to learn.
- As of now we plan to restrict the image to being made of straight lines after processing. Smooth curves will be approximated as being made up of small straight lines thus impacting the

accuracy of drawing. We can however try to use the differential mechanism of the wheels to attempt drawing them more accurately.

- We plan to develop a graphical interface for the user to specify the image to be drawn. However we have no idea on how to do this. The failover will be to use the command line interface.
- We might not be able to make/use a linear actuator for lifting and putting down the marker. Failover will be to use a servo motor to rotate the marker by an angle so that the tip doesn't touch the drawing surface. However during the lifting and putting down process, it will leave a light mark.