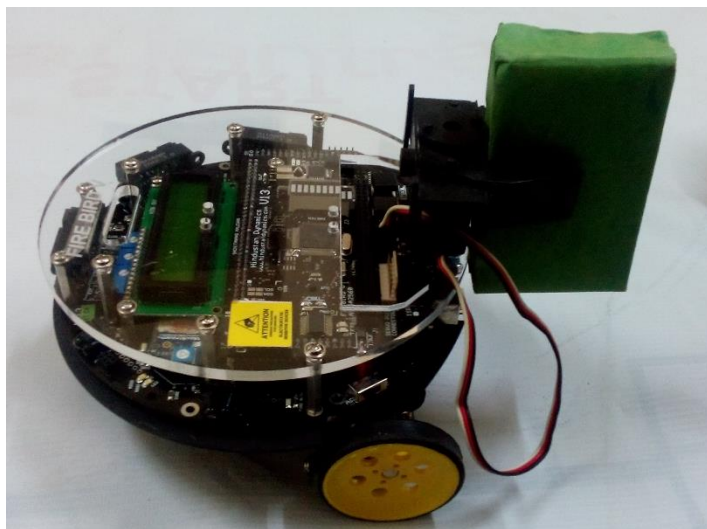


Software Project Report CS101

LIBRARY HELPMATE FIREBIRD V



TEAM ID-104

Saurabh Garg-140070003

Shaurya Arya-140070023

Pratik Brahma-14D070003

Abhin Shah-140070013

Contents

1. Project Overview

2. The Purpose of the Project

2.1 Background of the Project Effort

2.2 Goals of the Project

2.3 Measurement

3. The Scope of the Work

3.1 The Current Situation

3.2 The Context of the Work

3.3 Work Partitioning

4. Requirements

4.1 Hardware requirements

4.2 Software requirements

5. Configuring Code and Setting up of project

6. Test Cases

6.1 Choosing the designated path

6.2 Circumventing the obstacle

6.3 Line following

7. Software Architecture

8. Challenges and their Solutions

9. Design Constraints

10. Bugs

11. Future work

12. References

1. Project Overview

Development of library helpmate bot with the work of keeping unattended books at their designated location.

2. The Purpose of the Project

2.1 Background of the Project Effort

The problems faced by the book management staff in The Central Library and the students of IIT Bombay was the motivation for our project. Every library faces issues regarding wrong placing of books and, the labour and time involved in manual book management.

2.2 Goals of the Project

1. Develop a library database
2. Detect the color of a book using image processing and locate destination
3. Communicate with FIRE BIRD V using Xbee
4. Use line following for motion of bot
5. Detect and circumvent obstacles in the path
6. Use gripping mechanism to clasp books

2.3 Measurement

With proper implementation of this project, following benefits are envisaged-

1. Manual labour and time consumed of library staff will decrease by ninety percent.
2. Error in placing of books will be around one in thousand cases.

3. Instead of arranging books just once a day, we can have a dynamic placement of books throughout working hours.

3. The Scope of the Work

3.1 The Current Situation

The current status of library is faced with following problems-

1. After a book is being used, it usually stays unattended on the table for rest of the day.
2. Once a book is returned, it is kept at its place only at the specific time of book organization.

3.2 The Context of the Work

The intent of the project is to develop a system which interacts with its users, viz. library staff and students, in the form of library database development and input books to produce desired action. The bot interacts with the surroundings in the face of line following and avoiding obstacles.

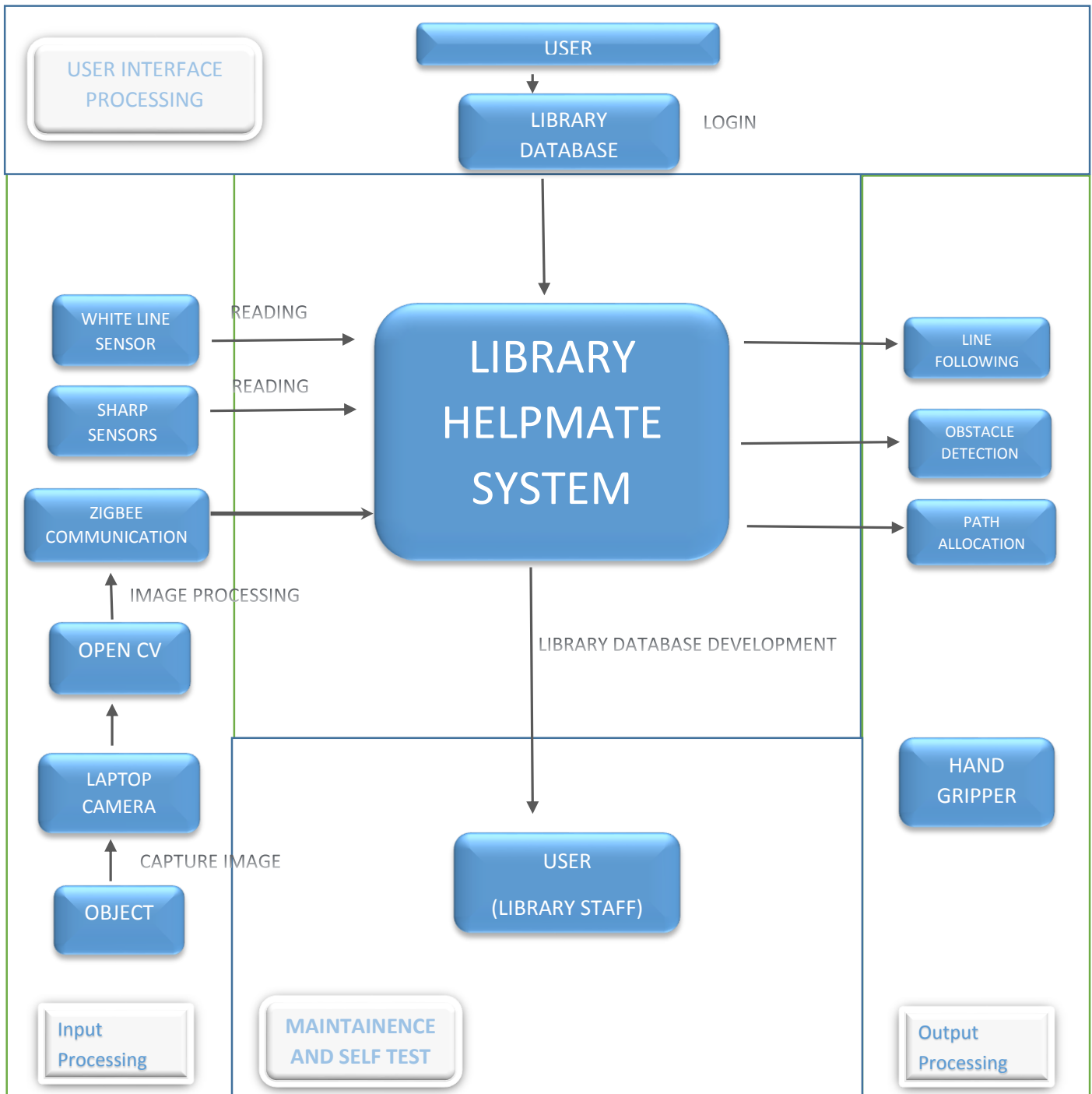


Fig 1 : Work context diagram

3.3 Work Partitioning

Following is the non-profit analogue of Volere's list of business events, which is effectively the same as a use-case diagram and a descriptive list of the associated use-cases.

Table 1 : List of Events

| Event Name | Input and output |
|--|--|
| 1. Book is identified | Image from webcam (in), Color of book(out) |
| 2.Bot clasps the book | Gripping action (out) |
| 3. Detected color is conveyed to the bot | Color of book to database(in) |
| 4. Bot follows line | Readings from IR sensors (in), line following action (out) |
| 5. Bot selects a path at the fork | First check (in) choosing correct path (out) |
| 6. Bot reaches destination and releases book | Destination check (in) book release, turnabout and path retraction (out) |
| 7. Ready for next book | Source check(in) Turnabout(out) |

Given below is a chart enumerating the functionalities of the Library Database.

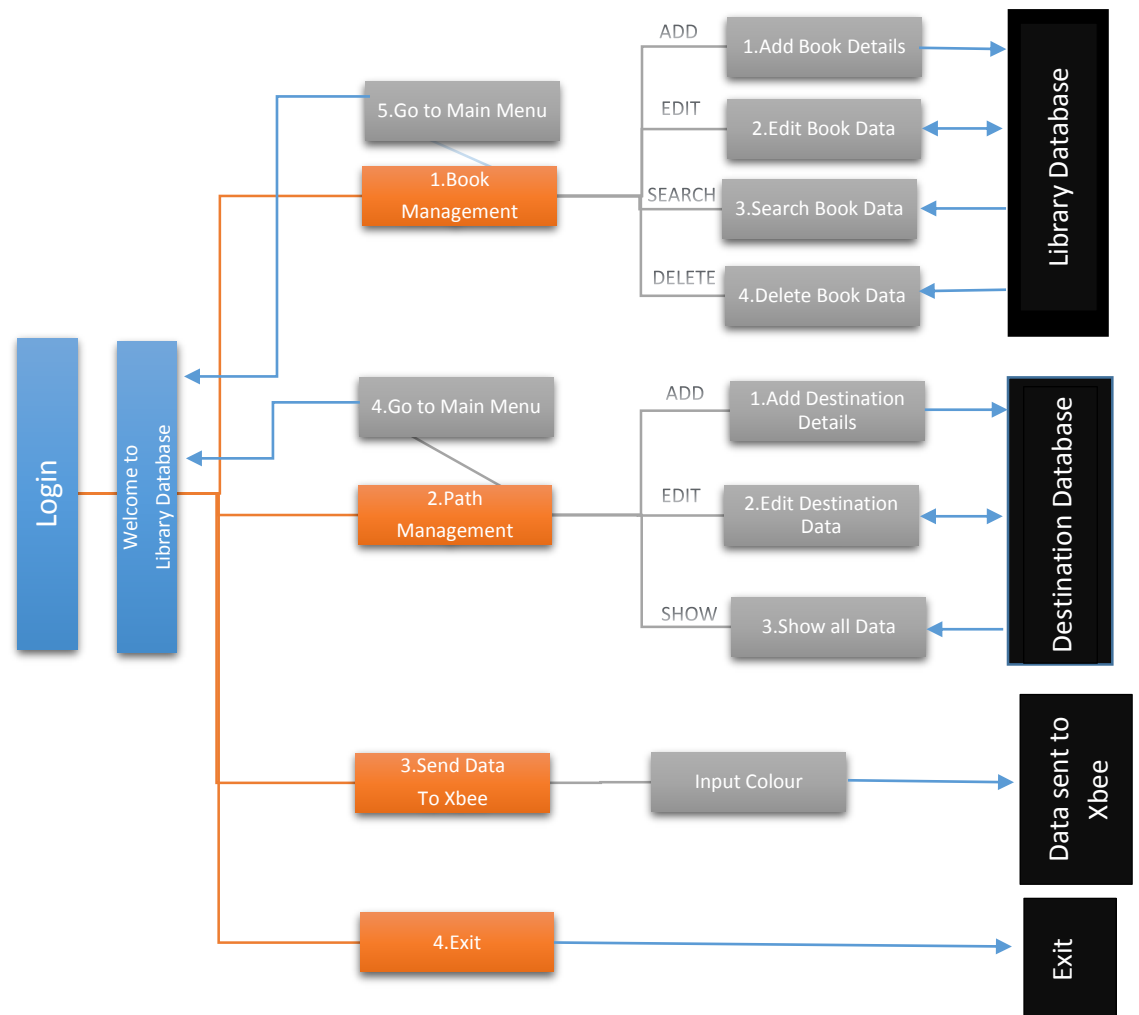


Fig 2 : Functionality chart

4. Requirements:

4.1 Hardware Requirements

1. FireBird V : Requires bot
2. WebCamera : To capture image of the book.
3. Zigbee : To communicate between bot and library database.
4. Sharp Sensor(3 required) : To detect the obstacle and circumvent them.
5. Servo Motor and Gripper: To hold the book.

4.2 Software Requirements

1. AVR studio: To program instruction onto a given bot
2. OpenCV: For processing images captured by camera
3. CodeBlocks: To run library database.
4. X-CTU: To configure Xbee modules.

5. Configuring Code and Setting up of Project

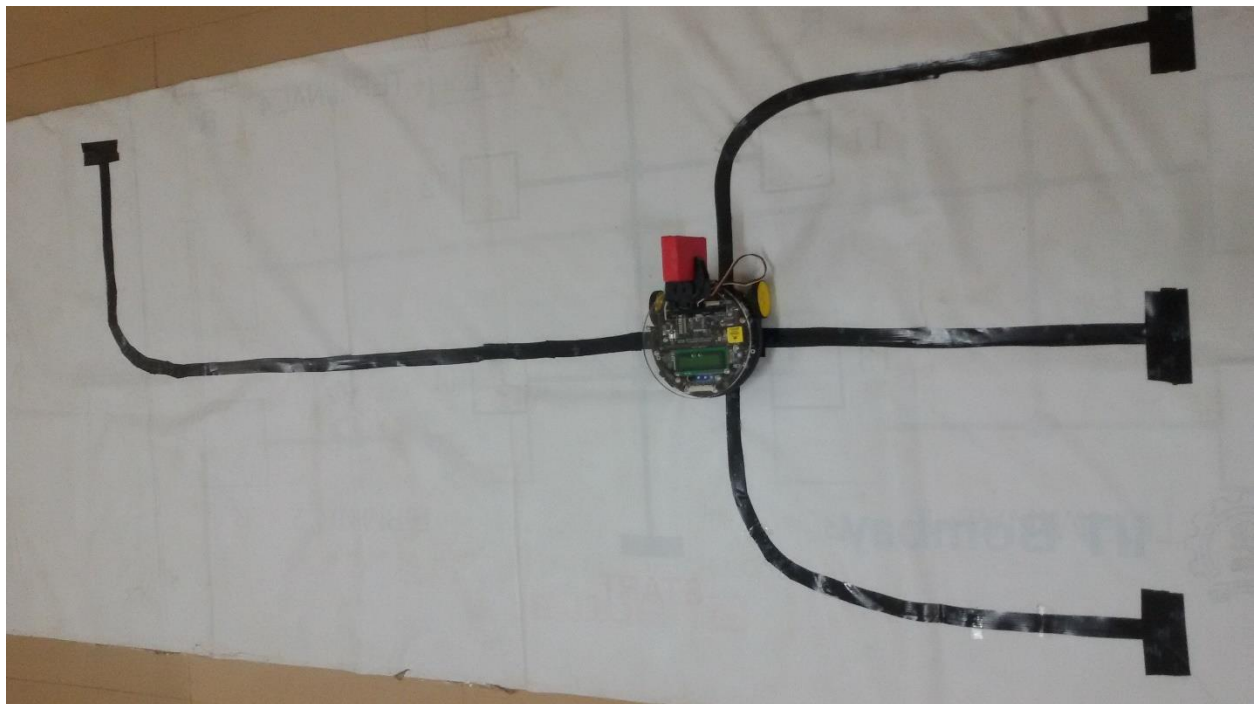
Instructions :

1. Follow the instructions at <http://kevinhughes.ca/tutorials/opencv-install-on-windows-with-codeblocks-and-mingw/> to install and integrate OpenCV with Code::Blocks.
2. Install X-CTU <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/xctu>

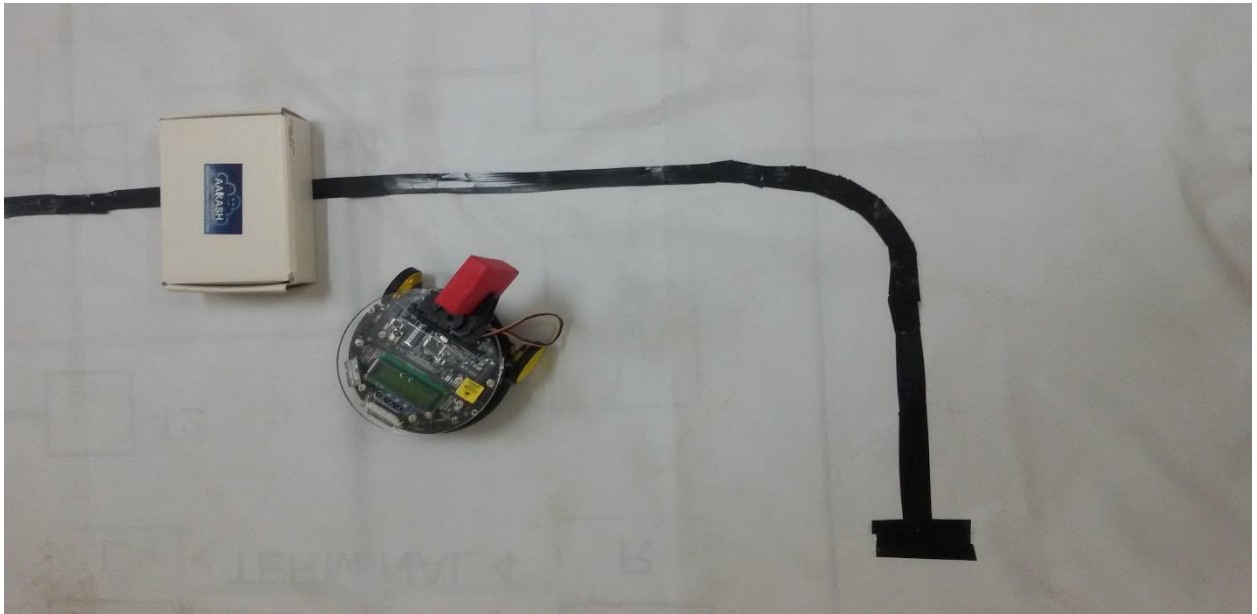
3. Open source code using Code::Blocks and, build and run Library_Database.cpp and Capture.cpp .
- 4 .Passkey to Library Database is "cs101".
5. Open Embedded_c.c in Atmel Studio, in Project column set the frequency as 14745600 in configuration option.
6. Build and run the Embedded_c.c code after including lcd.h header file to that directory.
7. Load the hex file generated in default option in that directory on Fire Bird V using AVR Bootloader.
8. Configure your Xbee using X-CTU. Refer to youtube link : <https://www.youtube.com/watch?v=Boy5asltEJw>

6. Test Cases:

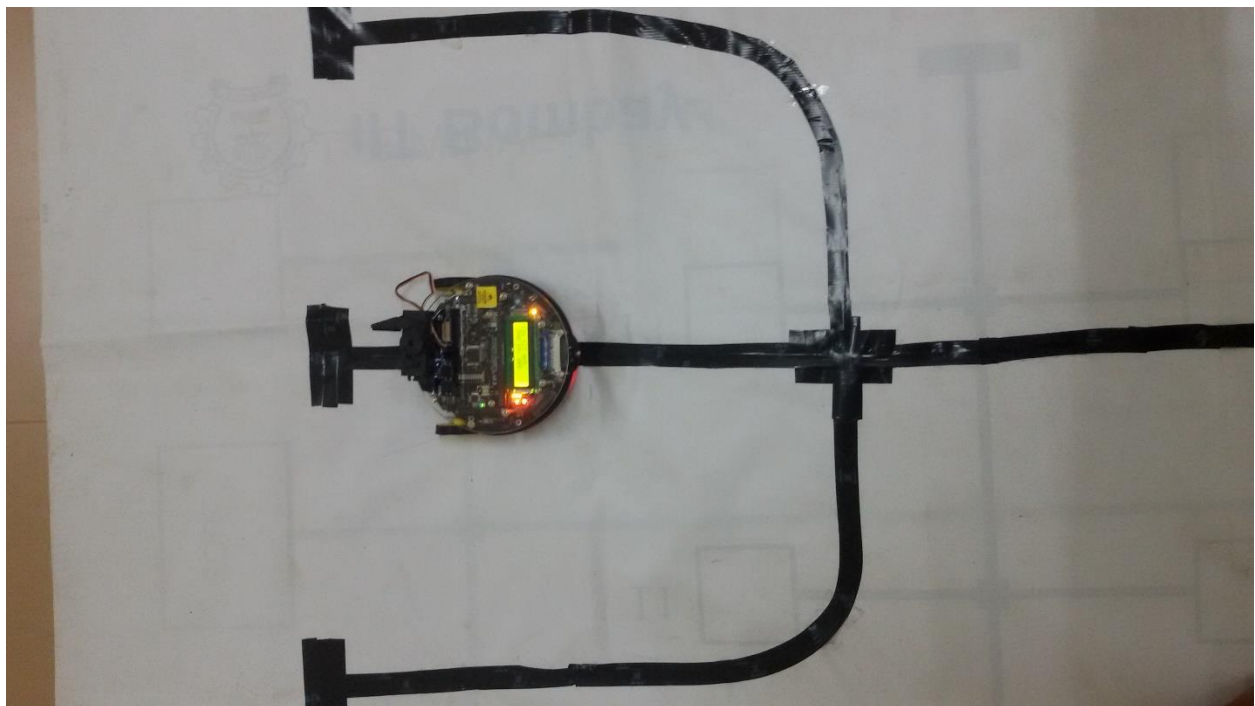
6.1. Choosing the designated path



6.2 .Circumventing the obstacle



6.3. Line following



7. Software Architecture

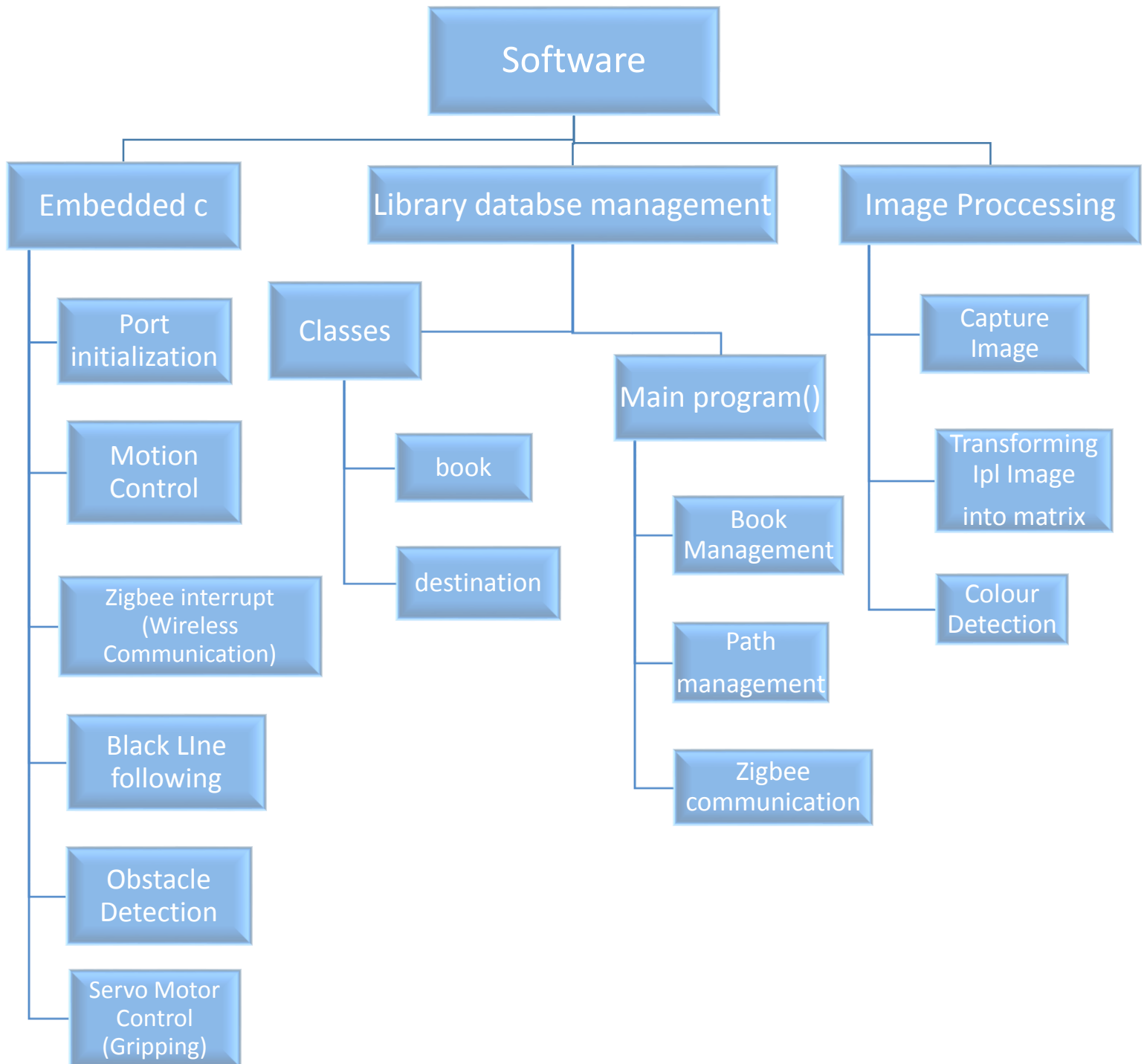


Fig 3 : Software Hierarchy

8. Challenges and their Solutions

Table 2 : List of challenges and ameliorations

| Challenge Name | Challenge Description | Amelioration |
|--|--|--|
| 1.Installing Open CV | Installing Open CV library using MinGW and C-make in Windows OS | Switched to Ubuntu |
| 2.Using XBee for wireless communication | Accessing port using libXbee | Inclusion of windows.h header file |
| 3.Line following | Inaccurate turning by the bot | Using Proportionality of previous errors |
| 4.Ignoring of checks by bot | Due to delay present in a function in the loop | Increasing thickness of line and checks |
| 5.Reading same check twice | Due to the backward motion included in line following | Increasing delay for forward movement after detecting the check |
| 6. Unable to detect exact color of the object | Due to inexact images captured by the camera . | Using better algorithm and limiting down to detection of only three color(RBG) |
| 7. Database | Creation of multiple copies of the file while adding in database | Adding the read function of the file in check condition of the while loop |

9. Design Constraints

1. For image processing book color should be either red, blue or green.
2. Line following path should be black line on white background or vice versa.
3. The range of the ZigBee module will determine the range of the user to control the bot within certain radius.

10. Bugs

1. Sometimes bot may move out of the path due to no sudden change in sensor values.

11. Future work

1. Better methods of book detection and expanding the range of colors from RGB to all the colors or, using barcode or QR code instead of colour as means for identification of book.
2. More Sophisticated line following by implementing complete PID (proportional-integral-derivative).
3. Automated hand mechanism so that the gripper can grasp the book on its own even from the ground.

12. References

1. *Firebird V Hardware and Software Manuals.*
2. *.ZigBee Reference Manual.*
3. <http://kevinhughes.ca/tutorials/opencv-install-on-windows-with-codeblocks-and-mingw/> to install and integrate OpenCV with Code::Blocks.
4. <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/xctu> to install X-CTU
5. <https://www.youtube.com/watch?v=Boy5asltEJw> to configure Xbee using X-CTU.
6. *A Sample Document for Generating Consistent Professional Reports- John T. Bell, University of Illinois Chicago.*

Git Repository :

https://github.com/saurabhgarg1996/140070003_104.git