THE UNIVERSITY of ADELAIDE

Software Design Document

for

Archaeology Robot
Group 13

Yufeng Bai 1600095	
Dawei Geng 1219181	
Jun Chen 1206265	
Quang Khoi Nguyen 1187070	
Shikai Li 1214223	
Yunyao Yao 1203525	
Yatong Zhou 1204471	

Version 0.1

September 8, 2012

Contents

Revision History

Name	Date	Reason For Changes	Version
Dawei Geng	03 Sep 2012	Template	0.1
Yufeng Bai	06 Sep 2012	Chapter 7	0.2
Yufeng Bai	06 Sep 2012	Chapter 8	0.3
Yufeng Bai	06 Sep 2012	Chapter 9	0.4
Yufeng Bai	07 Sep 2012	Fix the layout	0.5
Jun Chen	08 Sep 2012	Chapter 1	0.6
Jun Chen	08 Sep 2012	Chapter 2	0.7

Chapter 1

Introduction

1.1 Purpose and Scope

1.1.1 Purpose

The purpose of making this Software Design Document (SDD) is to give the details of the design of the archaeology robot and its system, which is designed by our group (group 13). In this document, we will give an overall description of architectural design, system organisation and critical modules of the system. This document describes some general ideas of how do we design the system and how do we implement requirements into the system. This document will be given to the team members so that they can construct the system based on the requirements. Also this document will be used as a reference for further developments.

1.1.2 Scope

This Software Design Document (SDD) is to give details of overall design of archaeology robot. It will contain nine parts : Introduction which will give an overall description of the design, System Overview which will show the design of the system, System Architecture and Components Design, Architectural Description which also includes alternatives and rationale, Data Design, Human Interface Design, Resource Estimates and Definitions, Acronyms and Abbreviations.

1.2 References

This Software Design Document (SDD) references these files below:

Software Requirements Document (SRS) : this document demonstrate all the requirements for this project.

Software Project Management Plan (SPMP) : this document shows the details of how do we manage our project.

UML files : the class diagram show how do we design and implement our system.

1.3 Overview

This Software Design Document will be organized in a way that describe System, Architectural Design, GUI Design and Resource Estimates logically. In the beginning of the document, it will give an introduction of this document (Purpose and Scope) and System Overview. After that, the document will give some high level architectural design of the whole system, which include System Architecture and Components Design, Architectural Description. Then the document will move to the Data Design, which also includes Design Details. After the Data Design, it will also shows the design of Human Interface and Resource Estimates. Lastly, there are some Definitions, Acronyms and Abbreviations at the end of the document. This document will follow the developing stages.

1.4 Constraints

The follow restrictions, limitations, and constraints will affect the design and implementation of the system:

- The robot should be set up correctly before testing.
- The robt can not be damaged.
- The robt can not wlak out of map.
- The robt can not push the wall and colliding is not permitted.
- The map should be set up correctly before testing
- The map is in fixed size (less than A1)
- There are some no-go zone on the map.
- The system must has save and load function.
- An acceptable latency is 300ms.
- There are some hidden wall in the map.

Chapter 2

System Overview

The main goal of the system is to let the robot can be used for survey an archaeological site containing the remnants of an ancient city on the premise that the robot can not be damaged.

Firstly, the system of the robot will use light sensor, touch sensor and ultrasonic sensor to locate some objects on the map, which include:

- walls
- hidden walls
- obstacles on the surface

Secondly, the robot which is used to survey an archaeological site is programmed by leJOS software. With this softeware we can design the functionalities of the robot correctly, so that it can detect every object on the map on the premise that the robot is not damaged. When the robot detect somethings, the system should deal with it by using a proper way which is described in the Sofeware Requirements Document (SRS).

Thirdly, the operating sofeware of the robot will have a nice graphic user interface (GUI) which contains:

- Control panel
- Map area
- robot information
- battery usage
- mode
- etc..

Lastly, the database of this project shall contain XML map files to store and update the survey map, the map shows :

- the whole map
- No-go zone
- detected walls
- detected hidden wall
- detected obstacles

- unexplored zone
- explored zone
- the boundary of the map
- initial point

In conclusion, the whole system of this project contains four parts:

- Client System which is installed in the robot for implementing its functionalities.
- User(Host) System which is used to control the robot and do things the clients want.
- GUI System is used to give a graphic user interface to the User System so that it is easier to be used.
- Database System which will store the survey result into a XML map.

With these four parts the system will work properly and do the task based on the requirements.

Chapter 3

System Architecture and Components Design

Chapter 4

Architectural Description

4.1 Component Decomposition Description

4.2 Detailed Components Design Description

4.3 Architectural Alternatives

4.4 Design Rationale

Chapter 5

Data Design

5.1 Database Description

5.2 Data Structures

Chapter 6

Design Details

6.1 Class Diagrams

6.2 State Diagrams

6.3 Interaction Diagrams

Chapter 7

Human Interface Design

7.1 Overview of the User Interface

The Graphical User Interface is used to communicate with the robot. The GUI is connected with robot, the client is able to control the robot by GUI. When the robot finishes its task, the client is allowed to switch off and disconnect with robot. In addition, the GUI demonstrates the status of the robot. The status includes the speed, the power and the location of the robot. The GUI also shows the obstacles, the "no-go" zone and hidden walls, which is able to be checked on the map window.

The GUI is able to demonstrate the follow functions for users:

- Save map to a XML file
- Load map from a XML file
- Change the control mode
- Demonstrate the Bluetooth and Battery status
- Control the moving direction of the robot(i.e. forward, backward, rotating)
- Demonstrate the Icon of the map
- Demonstrate the Log Information
- Change the speed of the robot
- Demonstrate the coordinate of the robot

The GUI is displayed below:

GUI.jpg

Figure 11: GUI

GUI2.jpg

Figure 12: GUI WITH MAP

7.2 Deatiled Design of the User Interface

7.2.1 Save map

The GUI includes a icon which is used to save the map from a XML file. When the client presses the File menu and choose the save option, there is a window jumping out, the client is able to choose the file where the client want to save.

Figure 13: SAVE MAP

7.2.2 Load map

The GUI includes a icon which is used to load the map from a XML file. When the client presses that icon, there is a window jumping out, the client is able to choose the map from there.

Figure 14: LOAD MAP

7.2.3 Change the control mode

The client is allowed to change the control mode by pressing the mode button. When the client presses the Auto mode button, there is a widget jumping out. Then if the client presses ”yes”, the robot changes to the Auto mode. The client is allowed to do the same operation to change to the Manual mode. The Log information display which mode it is currently.

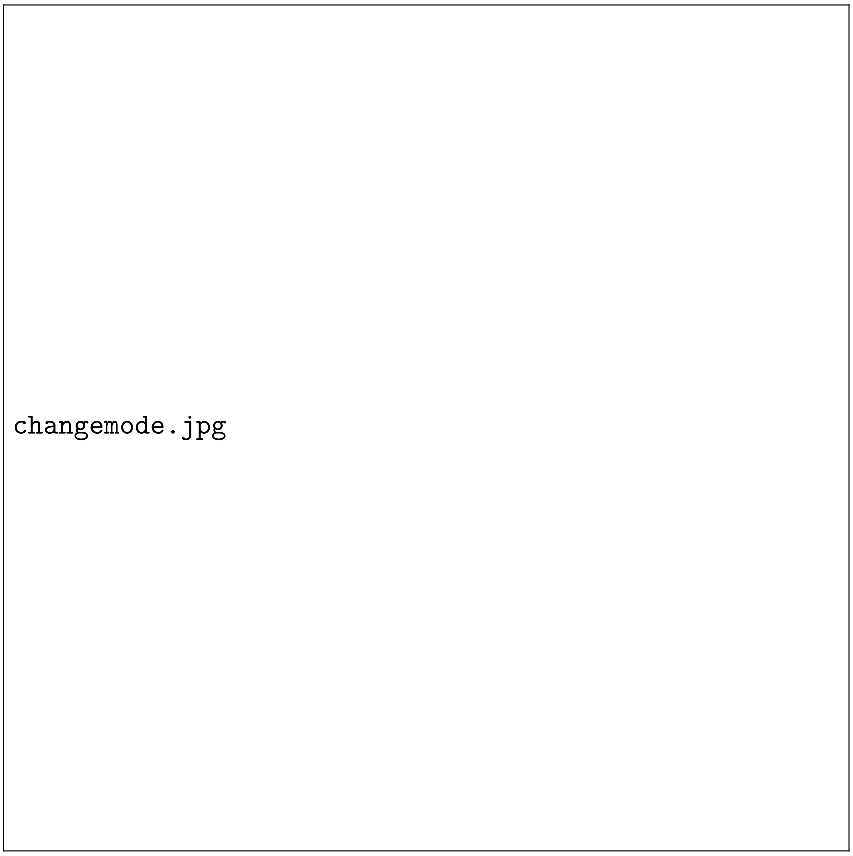

Figure 15: Change Mode

Figure 16: Change Mode icon

7.2.4 Demonstrate the Bluetooth and battery status

The Bluetooth and battery status are displayed on the GUI, if the bluetooth loses the connection, the log information board will demonstrate the lose information. The power of the battery is displayed by the percentage number and the colour of the battery bar will change along with the battery level.

Figure 17: Bluetooth and battery demonstration

7.2.5 Control the moving direction of the robot

The robot is able to be controlled by GUI when the robot is changed to the Manual mode. The client is allowed to press the arrows to move the robot. The robot is able to move forward, backward and rotate(include rotate 90 angles and rotate 360 angles). The moving direction controlling only can be used in Manual mode.

Figure 18: Moving control

7.2.6 Demonstrate the icon of the map

The robot and map information are required to be displayed on the GUI. It is necessary to use different icons to demonstrate the different map information.

Figure 19: Icon of the map

7.2.7 Demonstrate information of the GUI on the log information board

The Log Information board displays the information of the GUI. For example, the Log Information board demonstrates the version and the developers of the robot. If the client changes the control mode, the board displays which mode it is now. The board also is required to indicate the connection information.

Figure 20: The logging information

7.2.8 Control the speed of the robot

The GUI includes a speed bar and one specific number to control the speed of the robot. The client is allowed to change the speed of the robot by change the number of the bar. The larger number means the faster the robot is. If the number is zero, it means the robot stops immediately.

Figure 21: The speed controlling

7.2.9 The location of the robot

The GUI is required to display the specific location of the robot. The map is made up of a lot of grids. The location of the robot is able to display by coordinate, which is more accurate. When the robot moves, the coordinate will changes immediately.

Figure 22: The coordinate of the robot

Chapter 8

Resource Estimates

The resource of the project is estimated according to the Project Description and the client's requirements in every week's meeting. All hardware and software which are used in the project are demonstrated below.

8.1 Hardware

8.1.1 Robot

Name: NXT Robot includes all components of the robot
Function: The robot is used to search the map and find all the obstacles, hidden walls and "no-go" zone. All operations are required to finish by controlling the robot.

8.1.2 Host

Name: PC
Function: The PC is required to demonstrate the searching information of the robot. PC is also used to control the robot on Manual Mode. The programs are uploaded to the robot from the PC.

8.1.3 Connection

Name: Bluetooth and USB
Function: The Bluetooth and USB are all used to connect with the robot. The USB is limited by the length of the cable. The bluetooth is limited by the uploading speed. In this project, the USB is used to upload the program and the bluetooth is used to implement the manual control.

8.2 Software

8.2.1 Operation Environment

Name: Mac, Linux and Windows
Function: This project is not providing the working environment. Any system is able to develop the programs for the robot.

8.2.2 Developing language

Name: Java, latex
Function: The codes are allowed to write in java language, which is convenient to be read by any developers. The documents are required to write by Latex, then generating the pdf documents.

8.2.3 Developing tool

Name: Eclipse
Function: Eclipse is a better tool to make the program for this project. The Eclipse is used to write the operation commands for the robot and draw the GUI for this project.

8.2.4 Robot Software

Name: leJOS 0.9.1
Function: The leJOS is used to control the robot and developing tool is required to support the leJOS 0.9.1.

8.2.5 Testing tool

Name: JUnit
Function: The JUnit is used to debug the code.

Chapter 9

Definitions, Acronyms, and Abbreviations

9.1 Acronyms and Abbreviation

Acronyms/Abbreviation	Description
API	Application Programmable Interface
BT	Bluetooth
GUI	Graphical User Interface
PC	Personal Computer
SRS	Software Requirements Specifications
SPMP	Software Project Management Plan
UML	Unified Modeling Language
USB	Universal Serial Bus
XML	eXtensible Markup Language

Table 1: Acronyms/Abbreviations

9.2 Definitions

API: A set of classes and interfaces which are used to make the program for the robot include PC API and lejos API.

BT: A device to implement the wireless connection.

GUI: A interface which is generated on the PC is used to send the commands to the robot.

PC: A device which is used to make the program for the robot and build the GUI.

UML: A modelling language is used to build the diagrams for the project.

USB: A device to implement the wired connection.

XML: The data structure is used to store the map information.

Appendix A