*Automated Explorer And PathFinder*

project Management plan

Version *1.0*

*19/08/2016*

*GROUP 15*

*Ong Bee Lee*

*Foo Yuan Zhang Sean*

*Gede Bagus Bayu Pentium*

*Ng Yanrong, Lynette*

*Koh Wei Jun*

*Ng Jing Nee*

*Ren Daxuan*

*Tan Xuan Wei*

VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 1.0 | Ong Bee Lee | 19/08/2016 | Ong Bee Lee | 19/08/2016 | Filled some backbone details of the plan |
| 1.1 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

TABLE OF CONTENTS

[1 Introduction 4](#_Toc459379086)

[1.1 Project Objectives 4](#_Toc459379087)

[2 Scope 4](#_Toc459379088)

[2.1 Assumptions/Constraints 4](#_Toc459379089)

[2.2 Work Breakdown Structure 5](#_Toc459379090)

[3 Project Organisation 8](#_Toc459379091)

[4 Approach 9](#_Toc459379092)

[5 Risk Management 10](#_Toc459379093)

[6 Schedule/Time Management 10](#_Toc459379094)

[6.1 Milestones 10](#_Toc459379095)

[6.2 Project Schedule 10](#_Toc459379096)

[APPENDIX A: KEY TERMS 12](#_Toc459379097)

# 

# Introduction

## Project Objectives

This project aims to develop an autonomous robot which is able to:

1. Explore an arbitrary maze efficiently and create an accurate map of the maze, while avoiding any obstacles.
2. Determine the shortest path between the two given endpoints of the maze and traverse this path in the shortest amount of time.

# Scope

At the end of the project, the following should have been implemented:

1. A reliable communication interface between the multiple subsystems: Android Tablet, Raspberry Pi, Arduino (and PC).
2. A fully assembled robot, with appropriately calibrated sensors.
3. A GUI on the tablet, based on good Human-Computer Interface (HCI) practices, which will display the map of the maze, and controls to move the robot about the maze.
4. Software to calculate the shortest path between two points on a grid.
5. Software to control the motion of the robot, complete with error-correction to ensure minimal deviation in its motion (i.e., the robot is considered to have moved straight if it deviates less than 1° from its expected linear path).

## Assumptions/Constraints

The dimension of the arena measures 2.0m x 1.5m. Painted on the arena are grid corner markings, spaced 0.1m apart.

## 

## Work Breakdown Structure

Figure 1: Breakdown of the general tasks required to fulfill the project's requirements

| **No.** | **Work (Activity/Task)** | **Description** | **Efforts Estimation (manday)** | **Dependencies**  **(predecessor)** |
| --- | --- | --- | --- | --- |
| 1 | Setup of IDE | Setting up of the editors and IDEs to ease source code composition. | 2 |  |
| 2 | Assembly of Robot | Putting together the given scaffolding and the microcontroller kits. | 14 |  |
| 3 | Sensor Calibration | Generate a relation between sensor value and distance to obstacle for each sensor. | 7 | 2 |
| 4 | Motor Calibration | Ensure the pair of motors can drive the robot straight by implementing PID control. | 7 | 2 |
| 5 | Arduino Bluetooth Interface | Implement the code to communicate via Bluetooth. | 5 | 2 |
| 6 | Setup OS on RPi | Load up an OS (i.e. Raspbian) that can be booted on a RPi | 3 |  |
| 7 | Path Algorithm | Research on the available path search algorithms and adapt it to the project. Testing and simulation are done on MATLAB. | 21 | 6 |
| 8 | RPi Bluetooth Interface | Implement the code to communicate via Bluetooth on the RPi | 5 | 6 |
| 9 | Android GUI | Develop the interface with the arena map renderer and control buttons to remotely control the robot. | 12 |  |
| 10 | Android Bluetooth Interface | Implement the codes to communicate via Bluetooth on the Android application. | 5 |  |
| 11 | Android-RPi Communication | Be able to send and receive information between the RPi and Android application. Devise communication protocol. | 3.5 | 8,10 |
| 12 | Arduino-RPi Communication | Be able to send and receive information between the RPi and the Arduino. Devise communication protocol. | 3.5 | 5,8 |
| 13 | Milestone Review | Addressing the requirements stated in the given assessment checklist, and that our milestones have been achieved | (throughout project) |  |
| 14 | Debugging | Continual testing and troubleshooting of the various components of the system | 20 |  |
| 15 | Change Requests | Any issues that arise during the runs for leaderboard should be addressed as soon as possible | 3 |  |

# Project Organisation

|  |  |
| --- | --- |
| **Name** | **Role** |
| Ong Bee Lee | Project Manager  -Documenting the project and its progress  -Ensure that the project runs smoothly |
| Foo Yuan Zhang Sean | Arduino Developer  -Develop the program for the Arduino microcontroller to drive the robot motors.  -Integrate the Infrared sensors |
| Gede Bagus Bayu Pentium |
| Ng Yanrong, Lynette | Algorithm Team  -Research on the various path algorithms available  -Implement the algorithm with the RPi Developer to calculate the shortest path upon request |
| Ng Jing Nee |
| Koh Wei Jun | Android Developer  -Develop the Android tablet GUI as a remote controller for the robot |
| Tan Xuan Wei |
| Ren Daxuan | RPi Developer  -Install the required software packages into the RPi  -Implement the communication interfaces to communicate with the Arduino and the Android tablet via USB and Bluetooth respectively. |

# Approach

Our team has chosen to adopt the Agile Software Development Lifecycle Model (SLDC) as its model greatly reflects our planning process. We have divided up our goal into smaller, manageable tasks which are independent of one another to a great extent. Hence, each task can be developed and tested before being integrated with the rest of the parts. As the parts are developed, the merger is done iteratively, with further testing in between to ensure that everything falls into place with as little error as possible.

The bigger picture of our project can be summarized in the following chart:

Figure 2 Subsystem view of the project

**Arduino**

-Control motors

-Motion error correction

-Poll sensors regularly

**RPi**

-Path algorithm

-Dynamic map generation

**Android**

-Display map

-Remote controller

-Commands

-Remote control signals

-Sensor readings

-Remote control signals

-Map data

-Any feedback/error from robot

USB

Bluetooth

Autonomous Explorer and Pathfinder

Various IDE and editors will be used throughout the development based on the subsystem targeted.

* Arduino: Arduino IDE
* RPi: Pre-installed text editor
* Android: Android Studio/Eclipse

Simulation of our path-finding algorithm will be done on MATLAB to assess its efficiency and feasibility. Once the appropriate algorithm is chosen, it is ported over to C source code to be compiled on the Rpi.

# Risk Management

| **Risk Description** | **Mitigation/Contingency Plan** | **Criticality (Low/Medium/High)** |
| --- | --- | --- |
| Robot does not move straight | Implement PID error correction | High |
| Robot collides onto obstacle | Check for stray wires; revise algorithm | High |
| Robot does not move | Check the battery; Adjust the motor shield; Change power bank if needed | High |
| Path algorithm does not work as expected | Run tests; revise algorithm | High |
| Hardware faulty | Request for replacement by end of Week 4 | Medium |
| Rpi system crashes | Restart the system; check for short circuits | High |
| Rpi Bluetooth fails | Adjust Bluetooth dongle; Check the code | Low |
| Android GUI does not display the correct location of obstacles and robot | Recalibrate sensors; Revise the code | Medium |

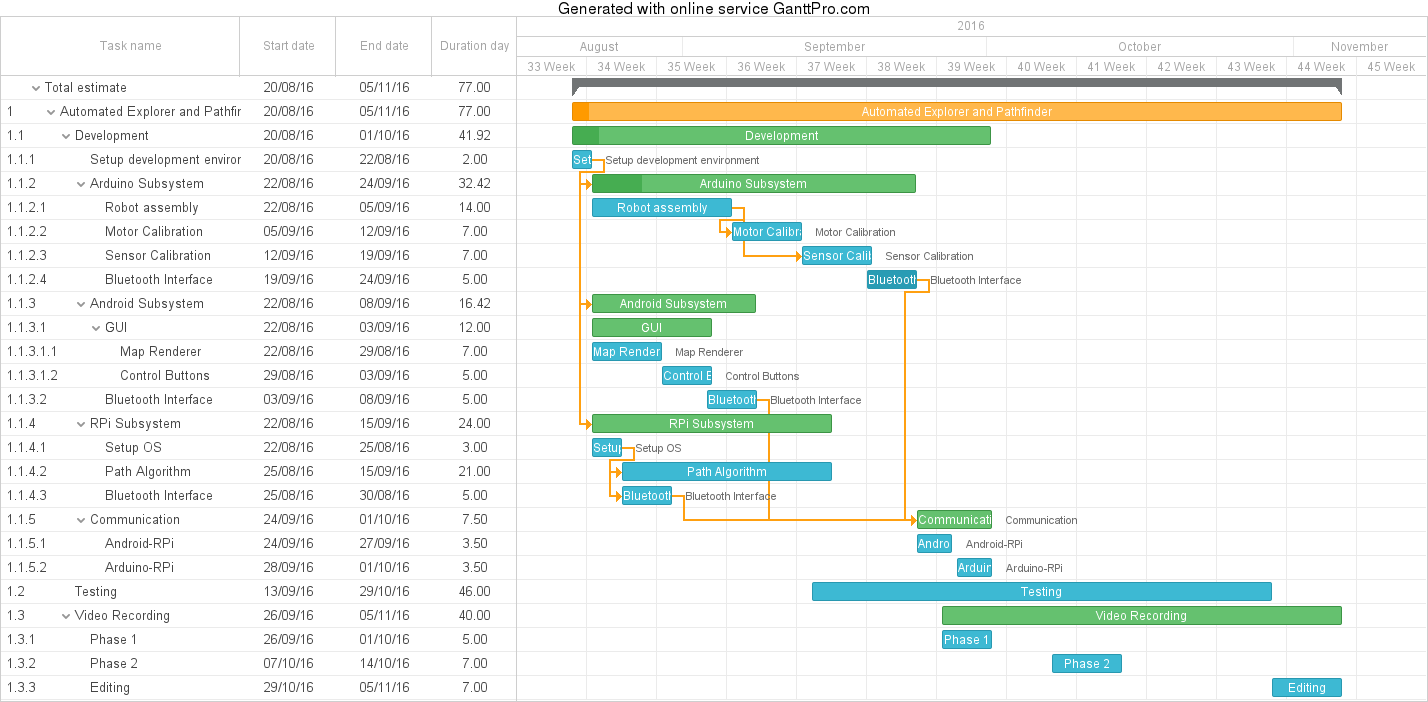
# Schedule/Time Management

## Milestones

The table below lists the milestones for this project, along with their estimated completion timeframe.

| **Milestones** | **Estimated Completion Timeframe** |
| --- | --- |
| Robot assembled | By end of Week 4 |
| Arduino subsystem completed | By end of Week 6 |
| Rpi subsystem completed | By end of Week 7 |
| Android subsystem completed | By end of Week 6 |
| Path algorithm | By end of Week 7 |
| Testing and Debugging | During Recess Week |
| Entered the Leaderboard | By the 1st attempt out of 5 attempts |
| Video Recording | By end of Week 12 |

## Project Schedule



APPENDIX A: KEY TERMS

The following table provides definitions for terms relevant to this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Arena/Maze | The playing field, on which the robot will be traversing. |
| GUI | Graphical User Interface |
| IDE | Integrated Development Environment |
| PC | Personal Computer |
| PID | proportional–integral–derivative controller |
| RPi | Raspberry Pi |