*Multi-Disciplinary Project*

PROJECT MANAGEMENT PLAN

Version *<1.0>*

*<31/01/2017>*

*GROUP04*

*Chia Sheng Jie*

*Tan Jie Ren Joel*

*Wan Maryam Binte Wan Mahmood*

*Tran Manh Tu*

*Tan Wanyi, Cherry*

*Yong Hui Hui*

*Koh Cheng Khim*

*Zhang Xinyi*

VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 0.1 | Tan Jie Ren Joel | 21/01/2017 | Tan Jie Ren Joel | 21/01/2107 | Initial Draft |
| 0.2 | Koh Cheng Khim | 21/01/2017 | Tan Jie Ren Joel | 21/01/2017 | Draft on WBS of Arduino programming part |
| 0.3 | Chia Sheng Jie | 28/01/2017 | Tan Jie Ren Joel | 28/01/2017 | Schedule Management |
| 0.4 | Tan Wanyi Cherry | 31/01/2017 | Tan Jie Ren Joel | 31/01/2017 | General Activities of Android |
| 0.5 | Yong Hui Hui | 31/01/2017 | Tan Jie Ren Joel | 31/01/2017 | General Activities of Android |
| 0.6 | Wan Maryam | 31/01/2017 | Tan Jie Ren Joel | 31/01/2017 | General Activities of Algorithm |
| 1.0 | Tan Jie Ren Joel | 31/01/2017 | Tan Jie Ren Joel | 31/01/2017 | Review of added parts and edits |

TABLE OF CONTENTS

[1 INTRODUCTION 4](#_Toc473721422)

[1.1 PROJECT OBJECTIVES 4](#_Toc473721423)

[2 SCOPE 4](#_Toc473721424)

[2.1 ASSUMPTIONS/CONSTRAINTS 4](#_Toc473721425)

[2.1.1 Assumptions 4](#_Toc473721426)

[2.1.2 Constraints 4](#_Toc473721427)

[2.2 WORK BREAKDOWN STRUCTURE 5](#_Toc473721428)

[2.2.1 General Activities 6](#_Toc473721429)

[2.2.2 Android 6](#_Toc473721430)

[2.2.3 Arduino 7](#_Toc473721431)

[2.2.4 Raspberry Pi 8](#_Toc473721432)

[2.2.5 Algorithm 9](#_Toc473721433)

[3 PROJECT ORGANISATION 9](#_Toc473721434)

[4 APPROACH 10](#_Toc473721435)

[5 RISK MANAGEMENT 10](#_Toc473721436)

[6 SCHEDULE/TIME MANAGEMENT 13](#_Toc473721437)

[6.1 MILESTONES 13](#_Toc473721438)

[6.2 PROJECT SCHEDULE 14](#_Toc473721439)

# INTRODUCTION

## PROJECT OBJECTIVES

The objective of this project is to build a robotic system that can autonomously explore and traverse an unknown area, avoiding obstacles. Such a robot will utilize algorithms and related hardware such as sensors to plan and follow a path from the start point to end point. The robot should be able to both explore and map out the entire area as well as plan and follow the fastest path from the start point to the end point.

# SCOPE

The following objectives are used to define this project’s scope:

1. Create a robot consisting of various components involving Arduino, Raspberry Pi,
2. Create an android application that controls the actions of the robot.
3. Design software algorithms that help define the process and logic that the robot takes to properly navigate the area.
4. Completed robot should able to move through an unknown area known as the Arena while avoiding obstacles that are laid out randomly. The Arena is a 2m x 1.5m rectangle, bounded by 15cm high walls and will contain obstacles that the robot must navigate around

## ASSUMPTIONS/CONSTRAINTS

### Assumptions

1. All team members are assumed to be able to fulfil their role despite some having limited knowledge and experiences about the various components prior to the start of project.
2. All essential hardware components provided are assumed to be compatible for integration.

### Constraints

1. The hardware components to be used in the robot system must be approved by NTU. Additional components other than those provided are not to be added on unless approved by NTU.
2. The sensors must be accurate and functional.
3. The wheels of the robot may degrade due to wear and tear, causing the robot to veer from the desired travel path.
4. The accuracy of the sensors and speed of the motors are unlikely to remain constant as the power fluctuates from consumption throughout testing.

## WORK BREAKDOWN STRUCTURE

C:\Users\Chia Sheng Jie\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Work Breakdown.png

### General Activities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Work (Activity/Task)** | **Description** | **Efforts Estimation (man-day)** | **Dependencies**  **(predecessor)** |
| A1 | Team organization and project initialisation | * Meeting of team members * Setting project’s goals, objectives and deliverables * Discussion on team responsibilities and team structure * Setting online sharing solutions (Google Drive, GitHub) for work distribution and efficient file sharing | 1 | NIL |
| A2 | Equipment testing and setup | * Setup necessary equipment such as Raspberry Pi, Arduino, Android and PC * Ensure all provided and required hardware are functioning properly. * Install necessary software for development | 3 | NIL |
| A3 | Initialize Project Plan | * Plan and delegate the tasks to be fulfilled in the project, map out project schedule | 2 | NIL |
| A4 | Update and Finalize Project Plan | * Update project plan as details and changes are added | 12 | A3 |
| A5 | Creation of group wiki | * Creation of group wiki to document findings and methods | 1 | NIL |
| A6 | Update and finalize group wiki | * Update wiki as details and changes are added | 50 | A5 |
| A7 | Video report | * Produce a video citing achievements and contributions in the form of a presentation for this project. | 7 | A6, B7, C5, D4, E6 |

### Android

The android application will act as the “steering” mechanism for the robot. It is responsible passing information from Raspberry Pi to the motors, which allows the robot to move. Android will also pass the information to an artificial intelligent agent.

The application will display the paths that have been explored by the robot in the form of a map.

The application will allow toggling between two modes, manual mode and auto mode. Manual mode enables full control of the on-screen controls to move the robot deliberately. Automatic mode enables the robot move based on the preset algorithm which determines dynamically the path it should take.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Work (Activity/Task)** | **Description** | **Efforts Estimation (man-day)** | **Dependencies**  **(predecessor)** |
| B1 | Functional Graphic User Interface (GUI) | * Develop the interface of the application to display the arena map along with the control buttons to enable the controlling of the movement of the robot. | 14 | NIL |
| B2 | Bluetooth Connectivity | * Implementation of codes to enable Bluetooth connectivity such that the android application is able to transmit and receive text. | 6 | NIL |
| B3 | Update Robot Start Coordinates | * To enable user to enter the starting coordinates for which the values will be transmitted via the Bluetooth serial link | 2 | B1, B2 |
| B4 | Display Current Status of Robot | * Indicates the status of the robot (i.e. stop, moving etc.) | 2 | B1, B2 |
| B5 | Display Arena Environment and Robot Location | * Display a map showing the known obstacles and the current location of the robot. | 2 | B4 |
| B6 | Manual/Auto Updating Display of Arena | * Manual Display Update: User manually updates the arena map * Auto Display Update: Arena map is automatically updated based on a regular time interval. | 7 | B5 |
| B7 | Testing of application | * Testing for the overall functionalities of the application. | 2 | B6 |

### Arduino and Hardware

In addition to the assembly of the robot, the Arduino team will also be responsible for the writing the program of the robot to achieve the functionality of sensors reading, motor driving and communication with Raspberry PI.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Work (Activity/Task)** | **Description** | **Efforts Estimation (man-day)** | **Dependencies**  **(predecessor)** |
| C1 | Functionality of Hardware | * Get hands-on experience and knowledge on all the given hardware parts | 3 | NIL |
| C2 | Calibration of Sensors | * Programming on Arduino * Generate the relationship between the distance to obstacle for each sensor and its sensor value. | 7 | C1 |
| C3 | Preliminary Design for the Robot | * Place the given hardware parts together in an organised and functional manner. | 3 | C1 |
| C4 | Movement of Robot | * Programming on Arduino * Ensure the robot walks in a straight line for at least 150 cm * Ensure the robot turns accurately based on specified angles (90 degrees etc.) | 14 | C2 |
| C5 | Repositioning and Recalibration of Robot | * Programming on Arduino * Organise the sensors and other hardware parts in their best position, making sure they are compatible with the algorithm of the software program. | 17 | C4, D4, E4 |

### Raspberry Pi

The Raspberry Pi will act as the “brain” of the robot, its job is to control the movement of the robot using a rational artificial intelligent agent. The artificial intelligent will be able to automatically explore the environment and find the shortest path towards a given goal. The Bluetooth interface serves as the manual remote control and map visualisation which is integrated with the Android Nexus 7 (N7). The Wi-Fi interface serves as the code data communication for the PC to transmit the data to the Arduino via Secure Shell Terminal.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Work (Activity/Task)** | **Description** | **Efforts Estimation (man-day)** | **Dependencies**  **(predecessor)** |
| D1 | Setup Raspian on RPI | * Reformat SD and Clean Install Raspian on the RPI. | 3 | NIL |
| D2 | Integrate Bluetooth between RPI and Android | * Implement the code to communicate between the Android and the RPI using Bluetooth. | 4 | D1 |
| D3 | Integrate Serial Connection between RPI and Arduino | * Implement the communication protocol to send and receive between the Arduino and the RPI using Serial Connection. | 4 | D1 |
| D4 | Establish Multi-Threading Coding | * Implement the code to establish bridging the executed code to control the robot via the path algorithm and at the same time, show the robot route path on the N7 GUI. | 7 | D2, D3 |

### Algorithm

The Algorithm team will be responsible for developing the robot’s exploration and movement algorithm to maximize efficiency and minimize the time required for completing its journey. The team will be responsible for translating the algorithm to the chosen programming language. The codes written will be ensured be well documented, readable and reusable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Work (Activity/Task)** | **Description** | **Efforts Estimation (man-day)** | **Dependencies**  **(predecessor)** |
| E1 | Preliminary Research | * Research on different algorithms, and decide which can be most efficient * Research on where it is most optimal to place the algorithm codes * Decide on a programming language and brush up on it if unfamiliar | 7 | NIL |
| E2 | Program Simulator | * Program simulator on PC to simulate the Arena | 1 | NIL |
| E3 | Path Algorithm | * Plan and implement an algorithm that can do a complete exploration of an unknown space * Plan and implement an algorithm that can compute fastest path from the start position to the goal position | 21 | E1 |
| E4 | Simulation Testing | * Simulate movement of robot on a grid map of the arena environment | 7 | E3 |
| E5 | Debugging and Physical Testing | * Consistent debugging of codes to minimize errors | 6 | E4, D4 |
| E6 | Optimization | * Improve algorithms to improve exploration and fastest path timings and minimize inconsistencies | 7 | E5 |

# PROJECT ORGANISATION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name** | **Matric No.** | **Primary Responsibility** | **Other Responsibilities** |
| 1 | Chia Sheng Jie | U1422474B | Project Manager | Raspberry Pi Communications |
| 2 | Tan Jie Ren Joel | U1422038A | Navigation Algorithm | Documentation |
| 3 | Wan Maryam Binte Wan Mahmood | U1422068F | Navigation Algorithm | Minute Taker |
| 4 | Tran Manh Tu | U1421423A | Navigation Algorithm | Algorithms Team Leader |
| 5 | Tan Wanyi, Cherry | U1521609J | Android Application | Android Team Leader |
| 6 | Yong Hui Hui | U1320018C | Android Application | Team Wiki |
| 7 | Koh Cheng Khim | U1520673K | Arduino Programming | Arduino Team Leader |
| 8 | Zhang Xinyi | U1421808J | Arduino Programming | Video Report |

# APPROACH

For this project, the development team will apply a modified waterfall model for the System Development Life Cycle (SDLC). This model is selected as the requirements of the project are already known upfront where the components for building the system, including the hardware requirements, software tools, and other necessary components are already established and are unlikely to change. However, there may be integration or conceptual errors made that may require the returning to the previous cycle after flaws were found downstream, especially as this team is attempting this type of project for the first time. Hence, a modified waterfall model is adopted to offer some flexibility. The modified waterfall model also allows the implementation of easy tasks in parallel with the difficult ones. In addition, a modified waterfall model offers the flexibility to revisit past process to make improvements and optimize the robot system such as implementing better algorithms, replacement of hardware components or code refinement. The goal is to minimize the exploration time of the robot system in the arena for both the Exploration test and the Fastest Path test.

With the modified waterfall model, various milestones will be established as benchmark goals but are still flexible enough to be modified as development progresses.

# RISK MANAGEMENT

The main purpose of Risk Management is to identify all kind of risks such as project risks that affect schedule or resources, product risks that affect the quality or performance of the software being developed, and business risks that affect the organization developing or procuring of the software.

The risks identified can be resolved/mitigated using the following strategies:

* Avoidance: To reduce the probability that the risk identified will occur.
* Minimization: To reduce the impact of the risk that has occurred.
* Contingency: To have a solution to resolve the risk that has arisen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Risk Description** | **Mitigation/Contingency Plan** | **Criticality (Low/Medium/High)** | **Probability (Low/Medium/High)** |
| 1 | Failure in integrating the codes from different software platforms. | **Avoidance Plan:** Research & identify potential integration issues that might arise beforehand and avoid unnecessary integration problems.  **Minimization Plan:** Use more widely available software that are known to be able to integrate into many software platforms.  **Contingency Plan:** Analyse the errors found during the integration of codes and solve these errors if possible. Change smaller parts of the system to other software platforms. In the worst-case scenario, be prepared to redesign the entire system. | High | High |
| 2 | Important hardware does not operate properly and achieve the required performance during testing. | **Avoidance Plan:** Do early testing of individual components of the robot and replace them if they are broken or faulty. Conduct weekly testing of components as well.  **Minimization Plan**: Separate development tasks to as atomic as possible so that broken hardware does not hinder all areas of progress.  **Contingency Plan**: Have spare parts ready for immediate replacement of components when possible to minimize time wasted during testing (e.g. having a spare battery). | High | High |
| 3 | Key member(s) are ill towards project submission deadlines. | **Avoidance Plan:** Encourage work-life balance and a healthy lifestyle through allocating a fair workload & not force members into situations where they spend excessively long hours on work without sufficient rest.  **Minimization Plan:** Have a buddy system amongst those with similar skillsets. Have them each understand his buddy’s work scope and be ready to take over when required.  **Contingency Plan:** Request help from other team members who are capable of assistance to complete the task as early as possible. | Medium | Low |
| 4 | The time needed to develop the software and algorithms is underestimated. | **Avoidance Plan:** Research the time based on similar project and complete the other areas of development for the robot early so that more time is available for algorithms testing.  **Minimization Plan:** Monitor progress carefully. Encourage visibility to allow the reviewing of progress and minimize delays due to underestimation.  **Contingency Plan:** Assign those from other teams early on to be prepared to assist the software and algorithms development team and have them consistently review the codes completed to be ready to step in to help. | Medium | Medium |
| 5 | Lack of skill needed to complete the task assigned to the team member. | **Avoidance Plan:** Interview team members to grasp an understanding of their strengths and weaknesses before assigning of roles.  **Minimization Plan:** Conduct reviews on team members’ work consistently as a team. This encourages visibility so that team members can help identify shortcomings of other member(s) and assist if required.  **Contingency Plan:** Encourage team members to request help from others who could be capable to assist to complete the task as early as possible. | Low | Medium |
| 6 | Lack of coordination between team members. | **Avoidance Plan:** Have team building sessions with the project manager helping to facilitate conversations and fun activities between the members early on. Ensure that each member clearly define the scope of their abilities to ensure that conflicts arising due to mismatched expectations are avoided.  **Minimization Plan:** Consistently monitor & review team member(s) work in short cycles. This helps ensure that low tension small changes can be made quickly without antagonizing another member.  **Contingency Plan:** Project manager to come in and resolve any conflicts between team members and improve their coordination through giving direct and precise instructions. | Medium | Medium |
| 7 | Restructuring of job responsibility for the project. | **Avoidance Plan:** Interview team members to grasp an understanding of their strengths and weaknesses before assigning of roles.  **Minimization Plan:** Project manager to prepare documentations on each job scope and ensure that everybody knows what is required on each role. At every phase of the development have each member review each document to familiarize themselves with it, regardless of whether it concerns them directly or not.  **Contingency Plan:** Project manager to actively delegate work and closely involve himself with all groups of team members to consistently check and ensure progress. This way he will be able to quickly devise a response and update the team in the event of changes. | Low | Low |

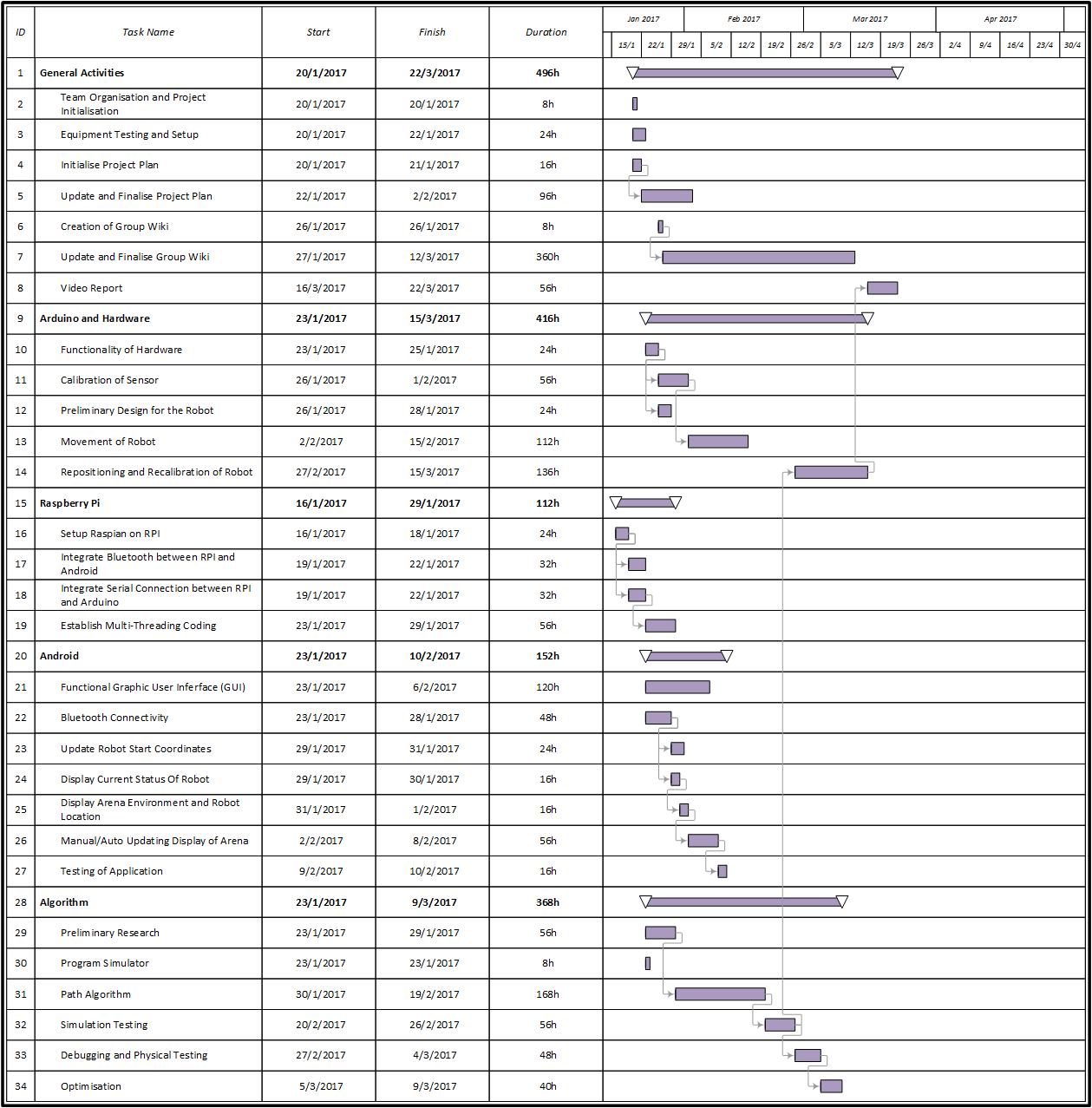
# SCHEDULE/TIME MANAGEMENT

## MILESTONES

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

|  |  |
| --- | --- |
| **Milestones** | **Estimated Completion Timeframe** |
| Allocation of the roles and responsibilities discussed for the project. | End of Week 1 |
| Documenting the Project Plan and Scheduling the future Wiki and Video plans. | End of Week 3 |
| Raspberry Pi Operating System Configuration and Wi-Fi/Bluetooth Integration. | End of Week 4 |
| Android Prototype with Remote Control and Map Location GUI completed. | End of Week 6 |
| Finalisation of Algorithm and Implementation of codes on the Robot. | End of Week 7 |
| Arduino and Hardware Fully Calibrated with Algorithm codes implemented. | End of Recess Week |
| Entering the Leader board for Both Exploration and Shortest Path Route. | By the 2nd attempt out of 5th attempts |
| Finalisation of the Checklist | End of Week 9 |
| Finalisation of Video Recordings | End of Week 12 |

## PROJECT SCHEDULE



**APPENDIX A: KEY TERMS**

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

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Arena | The playing field, on which the robot will be traversing. Dimensions: 2m x 1.5m rectangle, bounded by 15cm high walls and contains obstacles no be detected and navigated around by the robot. Also contains a starting point and “Goal” point. |
| GUI | Graphical User Interface |
| PC | Personal Computer |
| PID Controller | Proportional–Integral–Derivative Controller |
| RPI | Raspberry Pi Model 1 |
| Raspian | A Custom Debian Linux for RPI |