*MULTI-DISCIPLINARY PROJECT*

PROJECT MANAGEMENT PLAN

Version *1.8*

*04/02/2020*

*GROUP 01*

*SIEK MING JUN*

*DWIVEDEE LAKSHYAJEET*

*ADDI DEBASHREE*

*SU VOON HOU*

*TRAN HUU HOANG*

*MITTAL MADHAV*

*HOO BING YUAN*

*DENNIS CHRISTOPHER SUHERMAN*

*Supervisor*

*Loke Yuan Ren*

*Owen Newton Fernando*

*Tan Rui*

*Jwalapuram Prathyusha*

VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 1.0 | ADDI DEBASHREE & HOO BING YUAN | 22/01/20 | SIEK MING JUN | 22/01/20 | First Draft |
| 1.1 | SIEK MING JUN & DENNIS CHRISTOPHER SUHERMAN | 24/01/20 | SIEK MING JUN | 24/01/20 | Add work breakdown structure and project schedule. |
| 1.2 | DWIVEDEE LAKSHYAJEET | 27/01/20 | SIEK MING JUN | 27/01/20 | Add risk management, project organization. |
| 1.3 | MITTAL MADHAV | 28/01/20 | SIEK MING JUN | 28/01/20 | Add approach section. |
| 1.4 | SU VOON HOU & TRAN HUU HOANG | 30/01/20 | SIEK MING JUN | 30/01/20 | Update risk management, milestones and key terms. |
| 1.5 | SU VOON HOU | 01/02/20 | SIEK MING JUN | 02/02/20 | Added examples to constraints. |
| 1.6 | SU VOON HOU | 02/02/20 | SIEK MING JUN | 02/02/20 | Finalize the report. |
| 1.7 | TRAN HUU HOANG | 03/02/20 | SIEK MING JUN | 03/02/20 | Updated risk management. |
| 1.8 | SIEK MING JUN | 04/02/20 | SIEK MING JUN | 04/02/20 | Final check and modifications. |

TABLE OF CONTENTS

[1 INTRODUCTION 4](#_Toc31788637)

[1.1 PROJECT OBJECTIVES 4](#_Toc31788638)

[2 SCOPE 5](#_Toc31788639)

[2.1 ASSUMPTIONS/CONSTRAINTS 5](#_Toc31788640)

[2.1.1 Assumptions 5](#_Toc31788641)

[2.1.2 Constraints 5](#_Toc31788642)

[2.2 WORK BREAKDOWN STRUCTURE 5-9](#_Toc31788643)

[3 PROJECT ORGANISATION 10](#_Toc31788644)

[4 APPROACH 11](#_Toc31788645)

[5 RISK MANAGEMENT 11-13](#_Toc31788646)

[6 SCHEDULE/TIME MANAGEMENT 13-14](#_Toc31788647)

[6.1 MILESTONES 13](#_Toc31788648)

[6.2 PROJECT SCHEDULE 14](#_Toc31788649)

[7 Appendix a: Key terms 15](#_Toc31788651)

# INTRODUCTION

## PROJECT OBJECTIVES

The aim of the project is to develop and build a robotic system which is capable of exploring an unknown area, a maze of 2m x 1.5m x 15cm dimensions. Exploration is done autonomously, and the robot must avoid obstacles placed randomly on the arena. The robot must be able to plan and move through a path which is fastest from the starting point to the ending point of the maze through a predefined way point. The fastest route computed by the robot should avoid every obstacle by simulating the algorithm using simulators. The robot must be able to transmit and receive control signals from the mobile devices.

# SCOPE

The project involves building a robotic system that is comprised of 4 parts, namely Android, Arduino, Algorithms and Raspberry Pi.

**Android -** Design an application that is able to remotely control the robot and display the current location and status of the robot on the map by transmitting and receiving text strings over the Bluetooth serial communication link.

**Arduino -** Develop an Arduino Program that retrieves information from the sensors to send to the Raspberry Pi and calibration of the motors to ensure an accurate straight line motion as motors will generally exhibit slightly different motion characteristics given the same input signals.

**Algorithm -** Design the algorithm which enables the robot to explore the entirety of the maze and plan out the fastest route from the starting point and ending point of the maze through a predefined way point.

**Raspberry Pi -** Design and develop an interface responsible for the interconnection of various parts of the system (PC, Nexus 7 Tablet and Arduino) and support data exchange via Wi-Fi, Bluetooth and USB respectively.

## ASSUMPTIONS/CONSTRAINTS

## Assumptions

* Resources

The hardware provided by the lab is assumed to be sufficient for completing the task to build an autonomous robotic system to navigate through the arena.

* Manpower

It is assumed that all the members will carry out their allocated responsibilities sincerely with proper time management. This ensures an efficient team collaboration required for this project. Individual team members should meet the required deadlines for all deliverables to allow the team to complete the project on time.

## 

## Constraints

* Resources

The sensors provided have blind spots of 10cm, and calibration is required to correctly return the distance to obstacles.

The project has to be catered to be compatible with the provided software as some of them are of older versions, for example, the latest version of Python is 3.9.0 as of the creation of this document, 2020-02-01, but the latest version available on the RPi without having to compiling a newer Python version from source is 3.4.2.

Another example is OpenSSL. The OpenSSL version on available on the current Raspbian operating system is of version 1.0.1. As of the creation of this document, 2020-02-01, the latest version is 1.0.2. This may pose problems as Pip3 - Python 3.7.2 requires OpenSSL to be >= 1.0.2 for it to work properly.

* Limited Time

The robot is expected to be complete before week 8 for the leaderboard challenge and this might result in a stressful environment and a hectic project schedule. Hence, each team member is expected to manage time accordingly in order to deliver the project by the expected time.

## WORK BREAKDOWN STRUCTURE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Work (Activity/Task)** | **Description** | **Efforts Estimation (manday)** | **Dependencies**  **(predecessor)** |
| **Preliminary Project Governance** | | | | |
| 1 | High Level Project Discussion | * Meeting of team members. * Assign team’s responsibilities and team structure. * Set project’s goals, objectives & deliverables. * Set up Google Drive & GitHub for file sharing. * Discuss about project objectives and deliverables. * Read up & research on individual parts. | 2 days | - |
| 2 | Role assignment | * Assign roles based on strengths of teammates. | 1 day | 1 |
| **Project Planning** | | | | |
| 3 | Scope, Objectives and Assumptions | * Research on the scope and objectives of this project. * Make necessary assumptions if applicable. | 4 days | 2 |
| 4 | Project Plan and WBS | * Create a Work Breakdown Structure (WBS). * Plan out a project plan based on the WBS. | 4 days | 3 |
| 5 | Milestones, Project Risks and Key Terms | * Discuss and create milestones that we would achieve throughout the project. * Identity project risks that may occur. * Define key terms used in this documentation. | 4 days | 3 |
| 6 | Project Organization and Approach | * Create a project organization chart. * Discuss what are the approaches we are using for this project. | 4 days | 3 |
| 7 | Review and Sign-off | * Final review and sign off by Project Manager. | 1 day | 6 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 8 | Environment Setup | * Set up necessary environment needed for development. | 1 day | 7 |
| 9 | Analysis of Requirements | * Analyze & research up on requirements. * Find out how to achieve requirements. | 2 days | 8 |
| **Robot & System Design** | | | | |
| 10 | Knowledge Acquisition of Hardware Component | * Research up on robot’s system architecture and robot construction. | 2 days | 9 |
| 11 | Design of Robot System | * Design robot’s system architecture. | 3 days | 10 |
| 12 | Robot Construction | * Construct robot based on system architecture. | 3 days | 11 |
| **Arduino Subsystem** | | | | |
| 13 | Design Sensor Controller | * Research and design a controller for the sensors. | 4 days | 9 |
| 14 | Design Motor Controller | * Research and design a controller for the motor. | 4 days | 13 |
| 15 | Integrate Controller with Robot | * Integrate Arduino into robot. | 2 days | 14 |
| 16 | Testing & Refinement of Robot Movement | * Do sufficient testing of robot control based on Arduino’s controller. * Make necessary refinement if needed. | 2 days | 15 |
| 17 | Calibration of Sensors | * Calibrate sensors to match functional requirements. | 3 days | 16 |
| **Raspberry Pi (RPi) Subsystem** | | | | |
| 18 | Set Up & Familiarisation of RPi | * Research on RPi functionality and installation of Operating System and necessary drivers. | 2 days | 9 |
| 19 | Implement Serial Connection between RPi & Arduino | * Enable two-way communication between RPi and Arduino to relay instructions and data. | 3 days | 18 |
| 20 | Implement Bluetooth protocol for communication between RPi & Android | * Enable two-way communication between RPi and Android Tablet to relay instructions and data. | 3 days | 19 |
| 21 | Implement WiFi protocol for communication between RPi & PC | * Enable two-way communication between RPi and PC Host to relay instructions and data. | 3 days | 20 |
| 22 | Simulate communication between all subsystems | * Ensure robustness and interoperability of communication links. | 4 days | 21 |
| **Android Subsystem** | | | | |
| 23 | Design & Develop GUI | * Plan and Design a functional GUI in line with requirements. | 3 days | 9 |
| 24 | Implement Bluetooth Protocol | * Enable two-way communication between Android tablet and other robot subsystems in a reliable manner. | 4 days | 23 |
| 25 | Implement Robot Functional Requirements | * Design, implement and ensure desired robot functionality as per the checklist is reflected in the application. | 4 days | 24 |
| 26 | Implement Arena Functional Requirement | * Design, implement and ensure that arena-related visuals as per the checklist is reflected in the application. | 4 days | 25 |
| **Algorithm Subsystem** | | | | |
| 27 | Design and Develop Arena Exploration Simulator | * Conceptualize and build a realistic simulator of MDP arena for testing purposes. | 3 days | 9 |
| 28 | Design and Develop Fastest Path Computation Simulator | * Conceptualize and build a realistic simulator for the purposes of testing the fastest path algorithm. | 4 days | 27 |
| 29 | Generate Map Descriptor | * Design and implement file format based on map descriptor specification to fully encapsulate the state of the arena. | 4 days | 28 |
| 30 | Implement Time & Coverage-Limited Exploration Simulation | * Develop and showcase the operation of an algorithm in line with given constraints in terms of time and coverage. | 4 days | 29 |
| **Integration** | | | | |
| 31 | System Integration | * Consolidate all sub-systems in order to ensure interoperability. | 6 days | 12, 17, 22, 26, 30 |
| 32 | System Testing | * Perform extensive testing to ensure all systems are functional. | 4 days | 31 |
| 33 | Leaderboard Challenge – Trial Run | * Ensure robot conforms to the specifications and rules of the leaderboard challenge. | 4 days | 32 |
| 34 | Leaderboard Challenge – Actual | * Participate in leaderboard challenge during the stipulated period. | 20 days | 33 |
| 35 | Leaderboard Challenge – Refinement | * Make necessary adjustments based on leaderboard challenge feedback. | 20 days | 33 |
| 36 | Project Wiki | * Document technical specifications and learning process throughout the project. | 42 days | 9 |
| 37 | Final Video Report | * Showcase project achievements and learning points through a video report. | 3 days | 35 |

# PROJECT ORGANISATION

There will be two leads: hardware lead and software lead. Each will report to the project manager and will be responsible for a group of engineers as follows:

**Project Manager**

Siek Ming Jun

**Hardware Lead**

Dwivedee Lakshyajeet

**Software Lead**

Addi Debashree

**Raspberry Pi Engineer**

Su Voon Hou

**Raspberry Pi Engineer**

Tran Huu Hoang

**Arduino Engineer**

Mittal Madhav

**Algorithm Engineer**

Hoo Bing Yuan

**Android Engineer**

Dennis Christopher

Suherman

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Matriculation No.** | **Primary Responsibility** | **Duties** |
| 1 | Siek Ming Jun | U1820369F | Project Manager | * Manage the progress of the project * Algorithms team leader |
| 2 | Dwivedee Lakshyajeet | U1822289L | Hardware Lead | * Manage the hardware team * Team wiki editor. |
| 3 | Addi Debashree | U1722837C | Software Lead | * Manage the software team * Android programming |
| 4 | Su Voon Hou | U1720698D | RPi Engineer | * Meeting minutes taker * RPi communications programming * Image recognition development |
| 5 | Tran Huu Hoang | U1720201J | RPi Engineer | * Documentation * RPi communications programming * Image recognition development |
| 6 | Mittal Madhav | U1822408H | Arduino Engineer | * Developing navigation algorithms * Developing PC simulator |
| 7 | Hoo Bing Yuan | U1823670H | Algorithm Engineer | * Developing algorithms * PC simulator |
| 8 | Dennis Christopher Suherman | U1722583E | Android Engineer | * Developing the Android remote controller * Video report/editor |

# APPROACH

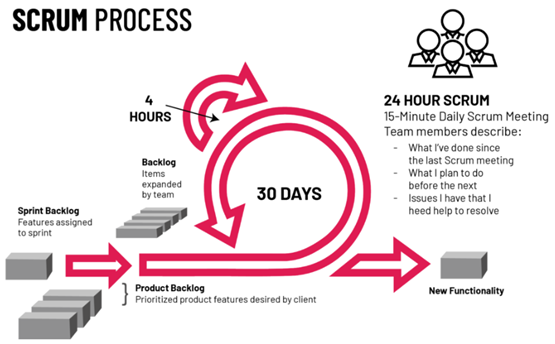


Figure 1: Scrum Model

For this project, the development team has decided to use the Scrum model as the primary SDLC methodology. Scrum is an Agile based project management framework and is used to manage complex software and product development using iterative and incremental practices. Scrum significantly increases the productivity of a team as compared to the classic “waterfall” processes. Scrum allows for the team to adapt to rapidly changing requirements in a smooth manner and produce a product that meets evolving goals.

In Scrum methodology, a sprint planning meeting is described in terms of the desired outcome (a commitment to a set of features to be developed in the next sprint). In a sprint, the team collaboratively sets their target with Product Owner as “Sprint Goal” and plan their work in “Sprint backlog”. The Scrum model suggests that projects progress via a series of sprints. These sprints usually last two – four weeks. For our project, each sprint will last one week.

As such, given the complexity of this project and the various responsibilities of the teams, the Scrum model will be the preferred SDLC methodology to use. It allows the teams to be kept updated weekly and allows for tight integration between the various teams. This prevent issues from occurring as teams will be able to resolve any conflict in the weekly sprint held. Weekly goals will be set during the meeting in order to ensure progress is made throughout the project.

# RISK MANAGEMENT

The potential risks can be mitigated using the following strategies:

* **Avoidance**: To minimize the probability that the risk identified will occur.
* **Resolution**: To have a solution to resolve the risk that has arisen.

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk Description** | **Mitigation/Contingency Plan** | **Criticality (Low/Medium/High)** | **Probability (Low/Medium/High)** |
| The team may be unaware of the progress of other members due to poor communication | **Avoidance**: All members will give updates regarding their progress on the WhatsApp group. A GitHub projects board will be used to give updates on the status of tasks.  **Resolution**: Analyze the progresses that needed to be speeded up in order for the whole project to run smoothly. Have other team member to work together on that part. Set the deadline for the progress to be finished. | High | Medium |
| Team members may not know what to work on | **Avoidance**: During the weekly scrum meeting, project manager will ensure each team member has assigned tasks and update the same on the GitHub projects board  **Resolution**: Issues can be raised on the GitHub repository and the person raising an issue can assign members to tasks. | High | Medium |
| Team members may not be motivated enough and procrastinate on tasks | **Avoidance**: Set up weekly target and progress plans to be met by individual team members for each scrum meeting.  **Resolution**: Leverage on individual peer review to discourage team members from procrastinating on tasks. | High | Medium |
| There may be a lack of programming knowledge | **Avoidance**: Before each scrum meeting, each member will be asked to research on the topics to be discussed during the meeting.  **Resolution**:Get other team member to help on specific problems. Encourage the member to practice on the domain they are lacking, and have them update the progress during the next scrum meetings. | High | Low |
| The team may encounter problems integrating the different components at the later stages of the project | **Avoidance**: Travis CI will be used for continuous integration on the GitHub repo. Before any code is uploaded, 2 members of the team will be required to peer review the code (from a different department).  **Resolution**: Set a stricter working schedule and shorter cycle of meeting, to get the project up and running before the deadline. | High | High |
| The tests for the robot may be insufficient | **Avoidance**: Before adding any feature, the team member will be required to propose at least 1 test. Each scrum meeting will also include discussions about the required tests for the scrum. | Medium | High |
| Some of the sensitive hardware devices may get damaged | **Avoidance**: The hardware will be handled carefully. Every week, all the components will be checked for proper functioning and any malfunction will be immediately reported.  **Resolution**: Ask for change of the hardware devices. If it is not allowed, responsible team member should afford usable devices. | Medium | Low |
| Team members may upload faulty code which may lead to the entire system malfunctioning | **Avoidance**: Eachpull request to the code base will require at least 2 peer reviews. Automated Continuous Integration (CI) tools will also be used to notify the uploader of any errors.  **Resolution**: Assign team member(s) to work on the faulty code. Have the team member(s) update the progress every meeting. | Medium | Low |
| The team may forget to track the progress of the project | **Avoidance**: The meeting minutes will be recorded for each scrum meeting.  **Resolution**: Have all team member to fill in the progress for their part(s) at the untracked meeting. | Low | High |

# 

# 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 roles and responsibilities. | End of week 1 |
| Documentation of Project Plan and Scheduling the future Wiki and Video plans. | End of week 3 |
| RPi Operating System (Raspbian) reformatting, configuration, and Wi-Fi/Bluetooth Integration. Submission of Project Plan. | End of week 4 |
| Completion of RPi camera configuration and image recognition algorithm. | End of week 5 |
| Completion Android Prototype with Remote control and map location GUI. | End of week 6 |
| Finalization of algorithm simulator and navigation algorithm. | End of week 7 |
| Implementation and integration of code on Robot/Hardware. | End of week 7 |
| Finalization of System functionality (Clear Checklist). | End of Recess Week |
| Arduino and Hardware fully calibrated with algorithm integrated. | By the end of 1st out 4 attempts (End of week 8) |
| Optimization of system performance (Algorithm, RPi, Image recognition, robot). | End of week 10 |
| Finalization of Project wiki. | End of week 10 |
| Finalization of report video. | End of week 11 |

## 

## 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. |
| CI | Continuous Integration |
| GUI | Graphical User Interface |
| PC | Personal Computer |
| RPi | Raspberry Pi3 and Enclosure |
| Raspbian | Raspbian (Jessie) is a Debian-based computer operating system for Raspberry Pi3. |