

CE/CZ3004 MULTIDISCIPLINARY DESIGH PROJECT

PROJECT MANAGEMENT PLAN

Version 1.0

31st August 2017

**Group 11**

Chen Ziao (PM)

Low Zi Qing

Sam Yong Shyang

Shafiqah Binte Rostam

Soh Jun Hao

Soh Koh Weng

Teo Jing Wen

Tham Guo Bin

Supervised By -

Li Fang

Sharad Sinha

Suresh Sundaram

Zhang Yaqian

*School of Computer Science and Engineering, Nanyang Technological University*

VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Implemented By | Revision Date | Approved By | Approval Date | Reason |
| 0.10 | Chen Ziao | 22/08/2017 | Chen Ziao | 22/08/2017 | Initial Project Plan Outline |
| 0.20 | Chen Ziao | 24/08/2017 | Chen Ziao | 24/08/2017 | Added Project Objectives and Scope |
| 0.30 | Soh Jun Hao | 26/08/2017 | Chen Ziao | 26/08/2017 | Updated Project Scope |
| 0.40 | Sam Yong Shyang  Shafiqah Binte Rostam | 28/08/2017 | Chen Ziao | 28/08/2017 | Added Assumption and Constraints |
| 0.50 | Low Zi Qing  Soh Koh Weng | 29/08/2017 | Chen Ziao | 29/08/2017 | Added Project Organization and Roles & Responsibilities |
| 0.51 | Chen Ziao  Soh Jun Hao | 30/08/2017 | Chen Ziao | 30/08/2017 | Added Work Breakdown Structure and Milestones (Arduino Team) |
| 0.52 | Teo Jing Wen  Tham Guo Bin | 31/08/2017 | Chen Ziao | 31/08/2017 | Added Work Breakdown Structure and Milestones (Algorithm Team) |
| 0.53 | Low Zi Qing  Soh Koh Weng | 01/09/2017 | Chen Ziao | 01/09/2017 | Added Work Breakdown Structure and Milestones (Raspberry Pi Team) |
| 0.54 | Sam Yong Shyang  Shafiqah Binte Rostam | 02/09/2017 | Chen Ziao | 02/09/2017 | Added Work Breakdown Structure and Milestones (Android Team) |
| 0.60 | Teo Jing Wen | 03/09/2017 | Chen Ziao | 03/09/2017 | Added Approach |
| 0.61 | Tham Guo Bin | 03/09/2017 | Chen Ziao | 03/09/2017 | Updated Approach |
| 0.70 | Chen Ziao | 04/09/2017 | Chen Ziao | 04/09/2017 | Added Project Schedule |
| 0.71 | Soh Jun Hao | 05/09/2017 | Chen Ziao | 05/09/2017 | Updated Project Schedule |
| 0.80 | Low Zi Qing | 05/09/2017 | Chen Ziao | 05/09/2017 | Added Risk Management |
| 0.81 | Soh Koh Weng | 05/09/2017 | Chen Ziao | 05/09/2017 | Updated Risk Management |
| 0.90 | All Team Members | 06/09/2017 | Chen Ziao | 06/09/2017 | Added Key Terms (Appendix A) |
| 1.00 | Chen Ziao | 06/09/2017 | Chen Ziao | 06/09/2017 | Finalized the entire Project Management Plan |

**TABLE OF CONTENTS**

Table of Contents

[1. INTRODUCTION 4](#_Toc492024547)

[1.1 PROJECT OBJECTIVES 4](#_Toc492024548)

[2 SCOPE 4](#_Toc492024549)

[2.1 ASSUMPTIONS/CONSTRAINTS 5](#_Toc492024550)

[2.2 WORK BREAKDOWN STRUCTURE 6](#_Toc492024551)

[3 PROJECT ORGANISATION 9](#_Toc492024552)

[4 APPROACH 10](#_Toc492024553)

[5 RISK MANAGEMENT 11](#_Toc492024554)

[6 SCHEDULE/TIME MANAGEMENT 14](#_Toc492024555)

[6.1 MILESTONES 14](#_Toc492024556)

[6.2 PROJECT SCHEDULE 15](#_Toc492024557)

# 1. INTRODUCTION

## PROJECT OBJECTIVES

The objective of this project is to build a robotic system which is able to navigate through a 200cm x 150cm territory without colliding into any obstacles by learning the environment. The robot must be able to calculate the shortest path from the start point to the destination point while passing through a predefined waypoint. The robot must also be able to receive control signals from a mobile device. All these functionalities should be achieved by integrating 4 subsystems which are Arduino, Raspberry Pi, PC with Algorithms and an Android Tablet (Nexus 7).

This project also aims to provide a platform where Computer Science students and Computer Engineering students can collaborate and apply various skill sets. It gives high exposure on how to integrate software system and hardware system into a single robotic system.

# SCOPE

The scope of this project contains 6 major parts:

1. Exploration and Shortest Path Algorithms Implementation

There are two algorithms needed to be implemented, which are Exploration Algorithms to explore the whole arena and Shortest Path Algorithms to find the shortest path among start point, destination point and predefined waypoint. It can be put on the Raspberry Pi or a PC.

1. Android Mobile Application Implementation

The Android Mobile Application is required so that user can interactively control the robot by transmitting control signal to the robot from the mobile device (Nexus 7). It should provide some basic functionalities including a functional GUI to visualize arena, Bluetooth connection to Raspberry Pi, and buttons to control the robot.

1. Arduino Hardware Implementation

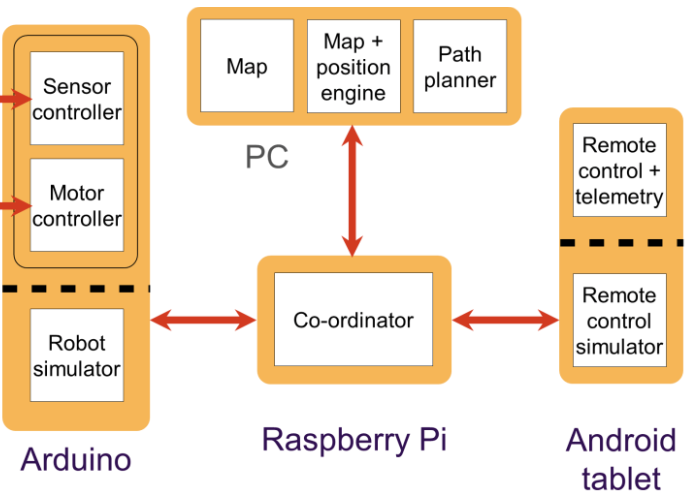
The Arduino hardware refers to the robot itself which consist of motors, sensors, batteries, cables and other hardware components. All components need to be soldered and wired properly. Calibrations needs to be performed to ensure the accuracy of robot movements.

1. Raspberry Pi Implementation

Raspberry Pi acts as the central CPU of the whole robotic system. With the Raspberry Pi 3 provided, a communication hub should be constructed, which enables Bluetooth, USB Wi-Fi and physical wiring.

1. Communication between the 4 parts above

Figure 1 below shows the connections between different components in the robotic system.



*Figure 1. Robotic System Architecture*

From Figure 1, It can be observed that there are in total 3 main communication channels. Connection between Raspberry Pi and Arduino is using physical wiring. Connection between PC and Raspberry Pi is using Wi-Fi. Last but not least, connection between Android Tablet and Raspberry Pi is through Bluetooth.

1. Documentations

Documentations such as project plan, project Wikis and project video are required to demonstrate the achievements attained through the projects. Documentations will be completed along with the robotic system implementation.

## ASSUMPTIONS/CONSTRAINTS

Assumptions:

1. Arduino Hardware components are in good working condition.
2. Android device (Nexus 7), PC and Raspberry Pi are able to provide reliable and stable communication.
3. Defective and missing components are replaced before commencement of development.
4. Power capacity from the supplied battery can last throughout each session of the leaderboard challenge.
5. The arena floor is assumed to be levelled.
6. Propagation delays from wireless connectivity and sensors are assumed to be negligible and will not significantly hinder the performance.

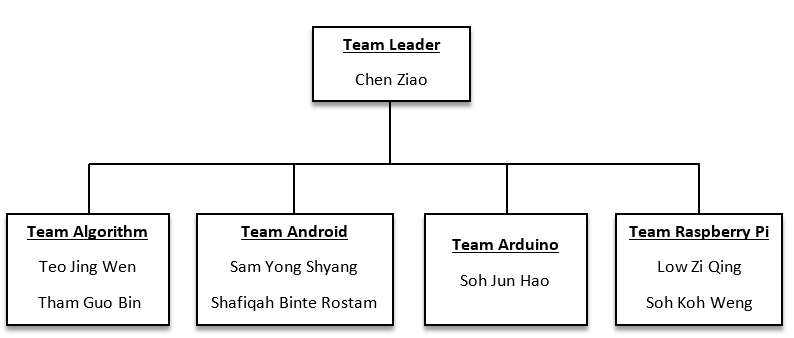
Constraints:

1. The power supply (batteries) cannot be changed to the ones with higher voltage supply.
2. Limited sensors (maximum 7) can be allocated.
3. There is limited range for wireless and Bluetooth connectivity.
4. Robotic system is subject to signal interference from other groups’ robots sharing the same wireless frequency, compromising the robot’s performance.
5. There may be loss and reestablishment of connection between mobile device and robot
6. The sensors do not give accurate measure of distance to obstacles. (It will fluctuate within certain range which may be large).

## WORK BREAKDOWN STRUCTURE

|  |  |  |  |
| --- | --- | --- | --- |
| **Work (Activity/Task)** | **Description** | **Effort Estimation** | **Dependencies** |
| **Project Planning and Management** | | | |
| Responsibility Allocation | Split the whole team into 4 groups which are responsible for different parts of the robotic system (Algorithm, Arduino, Android, Raspberry Pi). | 1 day | Not Applicable |
| Project Briefing (Algorithm, Arduino, Android, Raspberry Pi) | Members of 4 groups go to listen the respective briefings for the subsystem they are responsible for. | 1 day | Responsibility Allocation |
| Project Planning | Complete the project plan which includes project objective, scope, project organization, project schedule, approaches and risk management. | 14 days | Project Briefing (Algorithm, Arduino, Android, Raspberry Pi) |
| Project Wiki | Create and update an online website which contains project documentations. | 7 days | Project Planning |
| Project Video | Create a video to report achievements and contributions. The video should demonstrate creativity, presentation skills and teamwork. | 7 days | Final Leaderboard Competition |
| Team Meetups  (Once each week) | Team members meet up each week to give updates on respective achievement and discuss integration issue. | 9 days | Project Planning |
| **Algorithm** | | | |
| Learn and research | Understand the development environment and decide which programming language to be used. | 2 days | Not Applicable |
| Plan and design Arena exploration simulator program | Develop simulation environment and GUI of simulator program. | 7 days | Learning and research |
| Exploration algorithm | Design an algorithm to command the robot to the map completely and maps it out. | 5 days | Plan and design Arena exploration simulator program |
| Fastest path algorithm | Design an algorithm to command the robot to decide on the fastest path based on map from exploration from start to goal. | 5 days | Plan and design Arena exploration simulator program |
| Integration with Raspberry Pi | Integrate Arduino with Raspberry Pi 3 through Wi-Fi. | 1 days | Plan and design Arena exploration simulator program, Complete Raspberry Pi Implementation |
| Implementation/ Testing/Optimization of Algorithm | Testing the algorithm on the robot simulation program. | Ongoing | All tasks in algorithms |
| **Arduino** | | | |
| Assemble Robot | Assemble the Robot using all hardware components provided. | 2 days | Project Briefing (Arduino) |
| Research on Hardware Calibration Process | Learn how to calibrate hardware components such as sensors and motors through online resources and course materials. | 2 days | Project Briefing (Arduino) |
| Calibrate Hardware Components | Install software required and calibrate different hardware components to achieve desired outcome. | 7 days | Research on Hardware Calibration Process |
| Develop Arduino Application | Develop, maintain and modify Arduino application in an ongoing manner. | 5 days | Assemble Robot |
| Integration with Raspberry Pi | Integrate Arduino with Raspberry Pi 3 through physical wiring. | 1 days | Develop Arduino Application and Complete Raspberry Pi Implementation |
| **Android** | | | |
| Android app development preparations | Set up Android IDE, refresh Java knowledge, learn Android development basics. | 2 days | Not Applicable |
| Complete C.2 | GUI to initiate the scanning, selection and connection with a Bluetooth device. | 2 days | Not Applicable |
| Complete C.1 | Transmit and receive text strings over the Bluetooth serial communication link. | 2 days | Complete C.2 |
| Complete C.3 | GUI that provides interactive control of the robot’s movement via the Bluetooth link. | 2 days | Complete C.1 |
| Complete C.4 | GUI that indicates the current status of the robot. | 2 days | Complete C.1 |
| Complete C.5 | GUI to enter Waypoint & Robot Start Coordinates. | 2 days | Complete C.1 |
| Complete C.6 | 2D display of the maze environment and the robot’s location. | 3 days | Complete C.1 |
| Complete C.7 | Manual or Auto updating of maze display. | 2 days | Complete C.1 |
| Complete C.8 | Persistent and reconfigurable string command. | 1 day | Not Applicable |
| Complete C.9 | Robust connectivity with Bluetooth device. | 2 days | Complete C.2 |
| Integration | Bluetooth connection with Raspberry Pi, send commands to robot and receive information from robot. | 5 days | Complete C.1 - C.9 |
| Attempt C.10 | Extension beyond the basics, improve/innovate on existing application. (Open-ended) |  | Complete C.1 - C.9 |
| **Raspberry Pi** | | | |
| Configure Raspberry Pi | Install Raspbian OS on Raspberry Pi board and do the necessary configuration for SSH connection. | 1 day | Not Applicable |
| Develop Bluetooth Communication Interface | Develop the Bluetooth communication interface using Python. | 7days | Not Applicable |
| Develop USB Communication Interface | Develop the USB communication interface using Python. | 7days | Not Applicable |
| Develop Wireless Communication Interface | Develop the Wireless communication interface using Python | 7days | Not Applicable |
| Implement multithreading for the 3 interfaces | Implement multithreading for the 3 interfaces using Python | 7days | Develop Bluetooth Communication Interface complete, Develop USB Communication Interface  Complete, Develop Wireless Communication Interface  Complete |
|
|
|

# PROJECT ORGANISATION



*Figure 2. Team Organization Structure*

Multidisciplinary Design Project organizational structure is led by a project leader which main responsibility is to set a goal for the team and manage the administrative aspect of the team. Team leader focus will be coordinating between different sub groups within the whole team, ensuring that the deliverables are met and communication between teams are established.

The Multidisciplinary Design Project is made up of different sub teams which is responsible in its own field of specialization. Below are the main tasks of the sub teams:

1. Algorithm

Team members are responsible to construct an algorithm that is the most efficient in exploration and choosing the shortest path to the end point. With the implementation of additional requirement this year, the team would have to design the algorithm in a way that it must go through the way point before heading for the end point.

1. Android

Developing of an android application to input command via Nexus 7 Android Tablet to Raspberry Pi. Android application would have an overall view of the maze and the location of the robot. Android have to work closely with Raspberry Pi team to ensure that their work is synchronized in order to have a close integration process.

1. Arduino

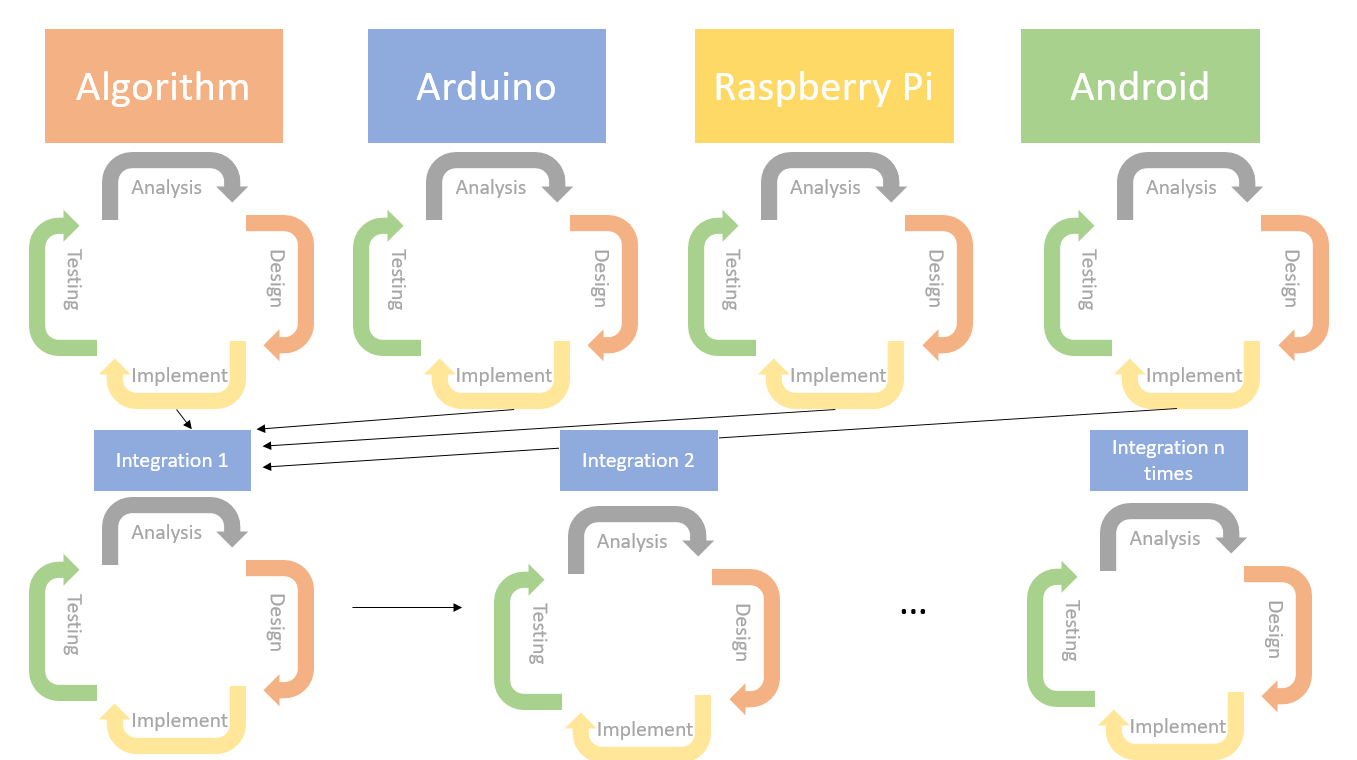
Responsible for the calibration of hardware components of the robot. The degree of turning of the robot is crucial, especially to achieve precision during exploration.

1. Raspberry Pi

Command and control for the overall project. The team is responsible to establish communication between all the different components, setup different protocols to ensure that the components can send message to one another. members will be working on Bluetooth, Wi-Fi, and multi-threading.

# APPROACH

**Iterative and Incremental Approach** is the chosen lifecycle framework that is integrated in this project. Each member (of the team) is tasked to a component of the project to develop and maintain. In consequence, each component of the robot system can be consistently refined by testing the latest iteration. This approach enforces the completion of a robot that is able to efficiently navigate through a territory.



*Figure 3: Project lifecycle framework: Iterative and Incremental Approach*

From the above diagram, our project has been split into 4 major components; Algorithm, Arduino, Raspberry Pi and Android which are headed by 2 person each. With every iterative cycle consisting of four major phases; Analysis -> Design -> Implementation -> Testing. Each phase will be developed and completed at different times but we will integrate each component once we have a working version of each component. Moving on, we will repeat the four phases after we integrate and keep incrementing to refine the robot system continuously.

# RISK MANAGEMENT

|  |  |  |  |
| --- | --- | --- | --- |
| Risk Description | Mitigation Plan | Criticality | Probability |
| **People** | | | |
| Members are unavailable for meetings | Avoidance Plan: The project leader should keep track of the availability of each member and plan the meeting dates ahead. | Low | Medium |
| Minimization Plan: One day before each meeting, project leader should check with each member and confirm their availability |
| Contingency Plan: Since the whole team is divided into 4 sub-teams, member of the corresponding team should inform the absent member with regards to any project updates. |
| Underestimated the time required to finish planned tasks | Avoidance Plan: Team members should plan their tasks with good estimation of time and effort. A reasonable amount of buffer time should be allocated to each task. | Medium | Medium |
| Minimization Plan: Members should update their progress on the current to the project leader bi-weekly so the team leader can access the progress. |
| Contingency Plan: Team should discuss any alternatives in completing the task. If not, project leader should modify the project plan accordingly by shortening time of other tasks, so that deadline can still be met. |
| Member do not have adequate skills to complete the team specific task | Avoidance Plan: Team members should attend all related briefing and do enough research on corresponding topics. | High | Low |
| Minimization Plan: Project leader should ask for updates from sub-teams frequently and ensure everyone understand their tasks. |
| Contingency Plan: Project leader propose a meeting with whole team to discuss the issue and brainstorm for solutions. If possible, project leader can restructure the team accordingly. |
| Inappropriate handling of hardware components leads to component failure and safety concerns | Avoidance Plan: Team members should learn relevant hardware specification materials carefully and understand how to handle them. | High | Low |
| Minimization Plan: When someone is handling the hardware, make sure that there is another member watching him and reminding him of any incorrect conduct. |
| Contingency Plan: Replace the broken hardware component and call for medical help if required. |
| Conflicts between different teams and members within each team | Avoidance Plan: Encourage more communication between different sub-teams and member within the team itself. Project leader should conduct weekly meetings to discuss issues and updates. | Medium | Low |
| Minimization Plan: Ensure everyone understand their tasks and have the same goal or sub-goals to achieve. |
| Contingency Plan: Conflicts should be reported to the project leader and discussed during team meeting. The final decision should be made by project leader after team discussion. |
| **Technology** | | | |
| Hardware failure (Arduino, motor, sensor, Raspberry Pi 3) | Avoidance Plan: Members in hardware team should read through all relevant materials about the hardware components and understand its usage and using conditions such as current limit and voltage limit. | Medium | High |
| Minimization Plan: The team members should handle the various equipment with care so as to not cause any damage to the equipment or injure themselves. |
| Contingency Plan: The whole team figures out any other alternatives to replace the components. Money to buy the new hardware should be deducted from the team budget. |
| Software failure (Android tablet, algorithms on PC) | Avoidance Plan: Members should design and develop robust and reliable software. | Medium | Medium |
| Minimization Plan: Before each leaderboard challenge, software check should start beforehand to ensure all software are working properly. |
| Contingency Plan: Carefully analyze the error and solve the bug without affecting other modules. |
| Whole robotic system integration failure | Avoidance Plan: Integrate different sub-systems one by one with intensive system testing. | High | Low |
| Minimization Plan: Do regular integration check before each leaderboard challenge. |
| Contingency Plan: Check for errors in all communication channels and locate the bug. Resolve the bug and do system testing again. |
| Defective equipment | Avoidance Plan: Sub-teams that are using any hardware components should test them and make sure all components function properly. | High | Medium |
| Minimization Plan: If not, members should approach lab supervisors and exchange any failed hardware components. Replace the faulty equipment as soon as possible by doing a one-on-one exchange (before week 4). |
| Contingency Plan: Purchase a new one from external shops (after week 4). |
| Data loss | Avoidance Plan: Team members are required to keep multiple copies of the source code to minimize loss of progress due to computer crashes. | High | Medium |
| Minimization Plan: Team members do a regular check on the data to ensure there is no data loss. |
| Contingency Plan: Use recovery tools to recover the data. If data cannot be recovered, team should decide on whether it is necessary to perform old tasks to obtain the data. If yes, team should modify the project plan according and start to recollect the data. |

# SCHEDULE/TIME MANAGEMENT

## MILESTONES

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

|  |  |
| --- | --- |
| **Milestones** | **Estimated Time Frame** |
| **Project Planning** | |
| Complete Project Plan | Week 4 |
| **Algorithm** | |
| Planning and designing of Arena exploration simulator program | Week 4 |
| Exploration algorithm | Week 6 |
| Fastest path algorithm | Week 6 |
| Implementation/ Testing | Week 7 |
| **Arduino** | |
| Assembled Robot | Week 4 |
| Sensor Calibrated | Week 6 |
| Motor Calibrated - Straight Line Motion | Week 6 |
| Motor Calibrated - Rotation | Week 7 |
| Arduino Program completed | Week 6 |
| **Android** | |
| Setup Android IDE Refresh Java knowledge Learn Android development basics | Week 3 |
| Complete C.1 & 2 - Transmit and receive text strings over the Bluetooth serial communication link - GUI to initiate the scanning, selection and connection with a Bluetooth device | Week 4 |
| Complete C.3 & 4 - GUI that provides interactive control of the robot’s movement via the Bluetooth link. - GUI that indicates the current status of the robot. | Week 5 |
| Complete C.5 & 6 - GUI to enter Waypoint & Robot Start Coordinates. - 2D display of the maze environment and the robot’s location. | Week 6 |
| Complete C.7, 8, 9 - Manual or Auto updating of maze display - Persistent and reconfigurable string command. - Robust connectivity with Bluetooth device. | Week 7 |
| Attempt C.10, extension beyond the basics | Recess Week |
| Integration | Recess Week |
| **Raspberry Pi** | |
| Configure Raspberry Pi | Week 3 |
| Bluetooth Interface complete | Week 5 |
| USB Interface complete | Week 5 |
| Wireless Interface complete | Week 5 |
| Multithreading complete | Week 6 |
| Integration | Week 7 |

## PROJECT SCHEDULE

**APPENDIX A: KEY TERMS**

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

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Exploration Algorithms** | A process to be followed to explore the unknown map within 6 minutes. |
| **Shortest Path Algorithms** | A process to be followed to find the shortest path from the start to end using explored map found. |
| **Arduino** | An open source 8-bit Micro-controller that control the robotic system |
| **Arena** | Physical Environment for the robot to maneuver and navigate around. |
| **Motors** | Pololu DV motor which runs at the rate of 120 RPM when 6V voltage is supplied. |
| **Sensors** | Infrared Sensor with Linear detector to detect & sense obstacle. Sharp IR sensor has different minimum distance (10-80cm & 20-150cm) |
| **Battery** | Battery to power the robotic system |
| **Android** | Mobile operation system that is developed by google which is based on Linux. |
| **Raspberry Pi** | The Raspberry Pi is a single board computer used in this project to facilitate the communication between the various components. (i.e. Arduino, Android Tablet, PC) |
| **Raspbian** | Raspbian is the Operating System of the Raspberry Pi |
| **GUI** | Graphical User Interface |
|  |  |