Group 9

Robot Software Development Project

Project Management Final Report

Version 1.0

Revision History

Date	Version	Description	Author
14/10/16	0.1	Drafted Sections 8 and 9	Mitchell T[redacted]
15/10/16	0.2	Drafted Sections 1 - 5	[redacted]
21/10/16	0.3	Completed sections 6 and 7 and added them to report	[redacted]
22/10/16	0.4	Proof-read and adjusted sections 6 and 7	[redacted] [redacted]
22/10/16	0.5	Proof-read and adjusted sections 1 - 5, 8 and 9	[redacted]
23/10/16	0.6	Integrated previous proof-read versions and adjusted sections 8 and 9	[redacted]
25/10/16	0.7	Proof read	Mitchell T[redacted] [redacted] [redacted] [redacted]
27/10/16	0.8	Completed Section 2 and Proof read	[redacted]
27/10/16	0.9	Slight grammar corrections	[redacted]
27/10/16	1.0	Made final adjustments and finalised document	Mitchell T[redacted] [redacted] [redacted] [redacted] [redacted]

Table of Contents

Project Management Final Report

1. Introduction

This final report of the Robot Software Development Project was completed by the team "Group 9" for the subject Project Management for Electrical Engineering (3620_ELEC_ENG_3024).

Group 9 consists of the following team members:

- Jason Tosolini
- Harrison Giannakis
- Mitchell Testen
- Lucas Sargent
- Brent Williams

1.1 Purpose

This document provides an in-depth analysis of the overall project. It is designed to give the viewer an insight into the effectiveness of essential aspects regarding the project and the lessons learnt during the development of the robot. The document is designed for the client and team to be able to review the project and the requirements that were met.

1.2 Scope

This document is generated as a summary and a review of the Robot Software Development Project. It details the changes made in the final product relative to the design documents generated in earlier stages of the project, as well as the knowledge obtained from making these adjustments. It also includes the technical solutions that were considered in making these adjustments.

1.3 Definitions, Acronyms, and Abbreviations

Robot – Refers to the UofA0015 Robot used in the project (see reference manual for details)

Software – Refers to the code written in C language to allow the Robot to operate

Hardware – Refers to the Robot itself (see Robot definition)

SoW – Statement of Work (provided by the Client)

SFD – Software Functional Description (provided by the Client)

SDTP – Software Design and Test Plan

1.4 References

[1] Manager, "PCX 009 TPL Project Management Final Report", PMEE, Adelaide, 2016.

[2] Manager, "3024.162.00-G-PE9-005 Statement of Work – Final Project Management Report", PMEE, Adelaide, 2016.

[3] Manager, XEMS, "EPC 001 EPR Document Version Control", PMEE, Adelaide, 2013.

[4] XEMS Manager, "PCX 005 CWI Numbering of Project Documents", PMEE, Adelaide, 2013.

[5] "Build software better, together", GitHub, 2016. [Online]. Available: http://github.com. [Accessed: 22- Oct- 2016].

[6] Project Management for Electrical Engineering, "Cost Estimation Lecture 6 - PMEE Combined Lecture Notes", pp. 239-245, University of Adelaide, 2016.

[7] D. Knight, S. Keen and B. Young, "UofA0015 Reference Manual", pp. 5-6, University of Adelaide, 2016.

2. Executive Summary

This document reviews the process that group 9 went through during the development of the specified software package, attempting to summarize the entire process. It describes the initial and recurring setbacks that were encountered, the problems that were overcome, and the lessons learned. It covers how the problem of concurrently running operations was approached at first and the issues that arose as well as how this was addressed and conquered. This document also includes major technical solutions to the project, reasons for choosing particular approaches - such as the stepping profiles for the motors - and clearly identifies how the product produced meets specifications.

3. Project Review

From the original plan, several project objectives were defined. The main objective was to produce a programmable robot that would satisfy the specifications of the client and move through a set of given coordinates. Excellent customer service was also to be provided to the client through fortnightly consultations in which any concerns could be addressed. This would allow the client to be aware of the progress of the project throughout the development process, thus keeping the customer satisfied. After completion of the main objective a document recording all necessary information about the robot was to be written that includes the progress and final specifications of the robot. This would give future programmers who may need to implement a similar design a detailed description of how this project was carried out and what design procedures were used during the development process.

Throughout the development of the robot many unexpected events occurred which forced the original plan of the project to be adjusted. While implementing the concurrent functions an issue was encountered, there was no solution to deal with these events within the original design. To solve the problem functions were consolidated to update each cycle. For example, to run the LED display in parallel with the motors the function to illuminate the LED display was called every 30 motor steps. Several weeks into the project the client requested a demonstration of the robot's basic functionality. This involved showing that the robot moved forward, backward, left, right and had the ability to display a string on the LED display. Leading up to the deadline of the final product the client required three additional feature to be added to the software package. The addition of these features was not possible with our current timing framework and required a redevelopment of the overall software structure. In conclusion, the overall project specifications were achieved within the required timeframe despite numerous setbacks.

4. Planning Effectiveness

4.1 Comprehensiveness of Planning Phase

A complete detailed plan of how the robot was to be implemented was constructed so that the team had a clear idea of what needed to be done. However, during the coding phase it was discovered by the team that the consideration of having concurrent events was overlooked, for this reason the planning phase needed to be re-evaluated so that the new design allowed for concurrent events. This conclusion was not reached until the software development stage was well underway and as a result the overall progress of the project was set back a significant amount. An oversight of this scale meant that the group had to revisit the planning stage and re-evaluate how the software could be restructured to account for this problem. This extended the time consumed during the planning phase, causing it to extend past the expected delivery date.

4.2 Scheduling, Resource Allocation and Project Evolution

To monitor progress a Gantt chart (see figure 1) was constructed during the planning stage, giving the client an overall view of the projected timeline of tasks that would be done throughout the project. At first the Gantt chart timeline provided reasonable time constraints, but as time progressed it became more noticeable that the timeline was not realistic. Several issues occurred throughout the software stage which caused the deadline of various tasks to be reconsidered. One of these issues was implementing a Global Timer to account for concurrent events. Originally the Global Timer seemed to provide a solution to the problem but as time progressed more issues arose. For this reason, the Global Timer was eliminated from the final design and frequent update functions were used to allow modules to run in parallel. This process forced changes to be made in every module of the software and thus the completion dates for particular modes were re-evaluated and extended.

Although the Gantt chart was not realistic it provided all the necessary milestones that needed to be completed throughout the scope of the project. Each section of the project was carefully scheduled through the team's judgment of their expected duration. Due to the level of experience the team had with such a project, estimating the timeline for each task was very difficult and may have been one of the main reasons why the timeline was unrealistic. The original timing of tasks was designed to be very optimistic, with the team expecting that certain stages would extend beyond their original deadlines. There was room within the timeline of the project to account for these extensions and the team was aware of this and able to make use of it when necessary.

As the team gained more experience working with the robot, a deeper knowledge was founded by different members of the group which encouraged the design to grow and change. Each team member accepted this change in the design as these new ideas improved important aspects such as modularity within the design that would make the software package easier for future programmers to implement.

5. Project Management Effectiveness

5.1 Communication Effectiveness

Communication was an essential aspect to the team throughout the course of the project, it played a major role in identifying and understanding problems that needed to be solved. Over the duration of the project the team found it easy to communicate with one another, this was because each individual member was open to the opinions of others. There was a desire to understand other team member's thoughts on an issue before reaching an important conclusion. This played a major role during client meetings, as it allowed each team member to gain a better understanding of the client's needs and provide exemplary customer service. As the project progressed, verbal communication skills amongst each member played a vital part during project development and client meetings. It allowed each member to effectively address any concerns they had with the project in a way that was understandable to both the client and the team itself. Confusion could reduce the client's trust of the team's ability to meet deadlines, which could lead to potential risks that would have large impacts on the project (e.g. implementing functions that do not satisfy the client's requirements).

5.2 Risk Management

Risks within the project were placed into three categories. These categories were defined as: risks caused by difficulty in estimation, risks based on assumptions made during the planning phase and unforeseen risks. During the project, all 3 types of risks were encountered and were handled through the same Risk Management technique (IAPM) by the team as shown below [6].

1. Identification
   It is important to make sure that the potential concern that has been raised is identified as a risk that has an impact on the project itself. Otherwise if the concern has no effect on the project then the concern itself is not a risk and can be ignored.

2. Analysis
   After identifying the risk, it is important to recognize the likelihood of that risk occurring and the impact it would have on the project. This will determine how much a particular risk would need to be monitored by the team throughout the project. For example, if the risk is of low likelihood but high impact, then higher levels of monitoring (step 4) need to be taken to reduce the likelihood of this risk occurring.

3. Planning
   Planning ways to minimize the risk will reduce the likelihood of any risk occurring. This in turn reduces the probability of having extra time spent within the project to remove the impact of these risks.

4. Monitoring
   Once a plan has been devised to mitigate the risk, the risk must be assessed on a regular basis to check whether the current planning techniques are reducing the risk and if not further planning needs to be done.

During the course of the project there were many unforeseen risks that occurred, one of these risks was not being able to implement the client's new requirements. These new specifications were presented unexpectedly as the original plan of the software package was developed based on the SFD provided at the beginning of the project. However, although the impact of the risk was big, the team dealt with the risk by using the Risk Management technique above. To mitigate the risk, new milestones were added to meet these new requirements and the Gantt chart was updated and assessed by the team on a regular basis.

5.3 Project Monitoring Mechanisms

Many monitoring mechanisms were used to evaluate particular aspects of the project. To ensure that all members of the team were contributing to the project, a graph on GitHub [5] which recorded the number of commits made to the project over time was constructed. The graph was beneficial because it showed at which stages throughout the project team members contributed to the software package and documentation. This allowed the team to remain constantly updated on what tasks team members had completed. However, there was also a very big disadvantage to using this mechanism as it could not tell the difference between a large commit and a small commit. For example, a commit that only edited one sentence within a report would have the same weighting as commit that edits a whole paragraph, meaning although the contribution of each team member could be assessed through commits to GitHub[5], the amount of work completed within these commits could not be evaluated effectively.

5.4 Progress Tracking and Reporting

Throughout the scope of the project a Gantt chart was used to monitor progress, it consisted of all stages and milestones of the project that needed to be completed. These project stages were represented as bars, where the left-hand side of each bar signified the expected start date of a stage and the right-hand side represented the expected finish date. All uncompleted stages were symbolized as blue bars to indicate to the team which parts of the project needed further development. After each stage was completed the bar corresponding to a complete stage was changed to green. The progress of each stage was indicated by a subsection that showed the current progress of that stage. If there were any stages that appeared to have an unrealistic start or end date the Gantt chart was updated so that these stages were rescheduled to a time frame that was reasonable for the team to complete. Any Milestones that required the delivery of documentation or the demonstration of particular module functions on the robot had specific dates that were not changed by the team unless the client adjusted their due date. These dates were represented as diamonds within the Gantt chart to make the team aware of when key components of the project were due. Progress throughout the project was reported to the client through client meetings. This made the client aware of the team's progress and eliminated any concerns in regards to the status of the overall project.

6. Effectiveness and Quality of Technical Solutions

The project was exclusively based on the development of software for the robot, no hardware modifications were made. As a result of this, all technical solutions and test regimes were completed with software development as the focus. Due to the nature of the requirements in the SFD, there were multiple technical solutions that had to be considered to deal with the robot's operations. The software was broken up to fit with the different hardware components of the robot. This was done by implementing code specifically for the operation of the motor, lights, buttons and display. Each of these components required their own technical solutions, and after applying these solutions further testing and problem solving was necessary to integrate these individual components together.

6.1 Motor - Choice of Stepping Type

In regards to the motor, the client's expectations were that the robot should be able to move between various points, in order to do this the two stepper motors had to be operating to allow the robot to move in various directions. The key decision that was made in designing this system was the type of stepping that would be used, as a stepper motor can incorporate different methods of operation. The various stepping types were narrowed down to full stepping and sinusoidal stepping as they were considered to be the two key candidates. Code was written for both types and testing was undertaken to observe the movement characteristics of the two different stepping methods. The goal of this testing was to identify which of the two types would be better suited for use in the final product.

The testing showed that there was an observable difference between the movement characteristics of the robot when using full-stepping compared to sinusoidal stepping. It was discovered that sinusoidal stepping produced smoother movement than full stepping, this can be attributed to how sinusoidal stepping using three times the number of steps to move the same distance (see figures 2 and 3). A major benefit of this was that sinusoidal stepping could be far more precise when it came to moving from one point to another as it had a greater ability to stop as close as possible to a particular coordinate. This was very beneficial as the final product was only allowed a small margin of error and the group decided that reducing the error as much as possible would be a priority. These two advantages clearly indicated that sinusoidal stepping was a better option for the final version of the software, however it's implementation would be significantly more complex. This is due to the sinusoidal stepping type having 12 different states in comparison to full-stepping having only 4, so it would require far more code to successfully implement. Ultimately it was decided that the benefits of using sinusoidal stepping outweighed the sole disadvantage and hence it was selected as the method of movement that would be used by the motors for the robot.

6.2 Execute Mode – Concurrent Display and Motor Operation

The problem of ensuring that the display was lit up with the correct information was considerably difficult to deal with. Initially there was a plan in place to develop a Global Timer and use it to allow the motors to move through their steps with a variable delay length between each step occurring. This Global Timer would also provide the means of producing the delay in the display module so that specific characters could be displayed during periods when the robot is moving. The implementation of this Global Timer proved to be very complicated, with many more problems arising as a result. While it could solve the issues of having the display and motor operating concurrently, it broke a large amount of code in other areas of the software. The decision was made to simply use the delay function to handle implementing delays in all parts of the software. This was coupled with the display update function being added into the motor functions themselves. Using an approach like this increased the complexity of the code and meant that the motor functions would now be reliant on the display functions and vice versa. The cost of using this approach was deemed to be acceptable in order to ensure that the particular requirement of concurrent motor and display operation was met.

6.3 Execute Mode – Storing Current Position and Orientation

The client required that the robot must be able to stop mid-movement when the event button is pressed, this occurs by detecting if the event button has been pressed in the motor movement function. The motor function code returns the number of steps which have occurred after it is called. If the event button is pressed in the motor function, it will return a smaller number of steps than expected and sets a flag to go into Select Mode. After the number of steps has been returned, the steps taken are fed into a function which calculates how much the robot has rotated/moved. This updates the current angle and current coordinate to align with the robot's current position and stores it in the current coordinate index before moving onto the next coordinate. After the position of the robot is updated, the robot exits Execute Mode and enters Select Mode.

6.4 Execute Mode – Calculation of Steps Required for Operation

To meet the client's requirements, the robot must be able to move through 15 different coordinates. This will be accomplished by instructing the motor to operate for a certain number of steps in a direction. The key to solving this problem was to create a set of functions that could take the robot's current position and its next destination and produce a number of steps. The complexity of this was that each function can only return a single variable, in this case it was a number of steps which is determined after working out an angle and distance. One function is used to take the current coordinate and next coordinate and identify the distance between them before converting it into an integer value of steps which is then output to the forward motor function. The other function is given the robot's current orientation (in degrees) along with its current position and next position and it determines which direction the robot should rotate and how many degrees of rotation are required. The choice of degrees over radians in this function was made due to it being easier to visualise and test. The final output is a positive or negative number of steps based on whether the robot must turn left or right. This output is then passed into the turn motor function.

This solution allowed for any number of coordinates to be entered into the robot as it only focuses on two at a time. That is, the robot's coordinate storage could be extended past 15 and these functions would still allow for the correct operations to be performed. The code design is also quite simple as there is a very small amount of inputs and outputs that allow for easy testing and error-finding.

6.5 Evaluation of Finished Product

One of the main goals of the software development portion of the project was to make sure that the code written would be easily understandable and adjustable. This was to ensure that if future changes or additions were to be made, it would be quite simple to identify what area of the code would be able to incorporate that particular change. Each group member contributed to the writing of the code and hence a variety of different coding styles were involved in the final product, but the whole group reviewed all parts of the code to allow each individual member to have a strong level of understanding of how each function operated. Comments were included throughout the code so that there are descriptions of what a particular function does and how it works. This process allowed the group to verify that the code produced was of a high enough standard that it would be maintainable and modifiable in the future.

6.6 Completeness of Final Product

The client's requirements are detailed in the SFD, this described the requirement of four different modes for the robot to operate in. These individual modes then had certain necessary operations for the buttons, display, lights and motor to perform. Each of these individual criteria is identified and explained in the table below.

Welcome Mode

Requirement	Detail	Status
Start in Welcome Mode	Upon robot power being turned on, the robot must start in Welcome Mode	Met
Display is operational	Display must cycle through each group members Student ID as well the group number (with an acceptable delay)	Met
Event Button starts Select Mode	If the Event Button is pressed while in Welcome Mode, the robot must transition into Select Mode	Met

Select Mode

Requirement	Detail	Status
Display "Select"	The display must be on and have the string "SELECT" displayed	Met
Buttons 1 enters Execute Mode	While in Select Mode, if Button 1 is pressed the robot will cycle through to Execute Mode	Met
Button 7 enters Program Mode	While in Select Mode, if Button 7 is pressed the robot will cycle through to Program Mode	Met

Program Mode

Requirement	Detail	Status
Buttons must perform particular actions	The buttons are used to enter coordinates Button 1 – decrement y coordinate Button 2 – increment y coordinate Button 3 – toggle between units of 1 and 10 Button 4 – decrement x coordinate Button 5 – increment x coordinate Button 6 – increment coordinate index Button 7 – Enter Select Mode	Met
Display must show certain characters	The display is used to show the current coordinates Display 1-3: shows y coordinate Display 4-6: shows x coordinate Display 7: shows coordinate index	Met
All coordinates start as null	Each coordinate is initially stored and shown on the display as (nn,nn)	Met
Start Position can be variable	The start position is initially (0,0) but can be changed to be any other coordinate	Met

Going outside the coordinate limits will revert coord to null	If the coordinate selected is going to be outside the -50 to 50 limits of the coordinate system then the coord shown revert to null (nn,nn)	Met
Exiting a null coordinate	If the current coordinate is a null coord then pressing inc or dec will cause it to revert to (0,0)	Met

Execute Mode

Requirement	Detail	Status
Display "Start"	Upon entering Execute Mode the string "START" must be displayed	Met
First Event Button Press	Pressing the Event Button initially causes the robot to begin its movement through the coordinates	Met
Display Operates while moving	While the robot is moving between separate coordinates the display should show a vertical line sweeping across horizontally	Met
Robot pauses on each coordinate	When the robot reaches a coordinate, is should pause for a moment and display the current coordinate in the same fashion as Program Mode	Met
Display "End"	Upon traversing all 15 coordinates the robot will display "END"	Met
Second Event Button Press	If the Event Button is pressed a second time while the robot is in motion, it will stop and enter Select Mode	Met
Remaining coordinates can be edited	If the robot is stopped during operation, the remaining coordinates must be changeable in Select Mode	Met
Re-entering Execute Mode will complete new cords	If the future coordinates are changed in Select Mode, re-entering Execute Mode should result in the robot moving to the new coordinates	Met

Robot must keep track of its current position and orientation	While the robot is operating in Execute Mode it should be aware of its current position on the grid and its current orientation (the direction it faces) so that it can continue operation after it has been stopped with a second press of the Event Button	Met

6.7 Testing Process

As the project was centred around developing software, testing was conducted at several different stages throughout the project period. This was consistent with the project method incorporating elements of the Agile method and having periods where code would be written, implemented, tested and then modified to improve its functionality. Each module of software involved numerous functions and the group endeavoured to thoroughly test each function as it was written. This meant that elements of the code that had been completed were verified to be working correctly, so that when other code was integrated with it, the location of errors could be easily identified and isolated. Using a function several times at different points in the software development process meant it would be tested multiple times and in an array of different scenarios. This allowed the group to ensure that the code was written to a high standard.

Reviewing the testing methods employed during the project has shown that the process used was of an acceptable standard as the code produced operated correctly and met the requirements of the client. The goal of the testing was to ensure that each section of the software could perform the task it had been designed for. As the result of the project was a software package that performed as required, it is fair to conclude that the testing method used was highly effective.

7. Project Deliverables

The deliverables for the client were as follows:

7.1 Project Plan

Due Date: 19/08/2016
Date Delivered: 19/08/2016

The Project Plan was submitted on the due date as required. Upon submission, all group members were confident that it met the standards of the client. Different sections of the plan were handled by different people, but all parts were completed in time for the whole group to review the document before committing to a final version. This process of proof-reading and analysing each component of the document was undertaken by the group as a form of quality assurance. Our goal was to ensure that we provided a Project Plan that was as accurate and detailed as possible, while ensuring that we met all the necessary criteria. The received feedback indicated that some sections were lacking in detail with certain points needing a more robust explanation. Overall, the document was completed to a reasonable standard.

7.2 Robot Movement Demonstration

Due Date: 31/08/2016
Date Delivered: 31/08/2016

The client asked for a small demonstration of the robot's capabilities to be made during the third meeting. The requirements were that the robot should be able to move in cardinal directions and display a string. Prior to the meeting, software development was divided up amongst group members, with the goal being to individually work on specific parts of the code that would later be integrated together to produce the final product. The operation of the robot, in regards to its ability to move and display a string, was thoroughly tested prior to the meeting to ensure that the quality of the work was up to the client's standards. The demonstration itself can be considered a success as the client was satisfied with the capabilities shown, hence this deliverable was finished to a high standard.

7.3 Software Design and Test Plan

Due Date: 16/09/2016
Date Delivered: 16/09/2016

The second major piece of documentation that was delivered to the client was the Software Design and Test Plan. This deliverable was submitted on the due date after going through numerous revisions as group members completed and evaluated each different section. The quality assurance process involved was very similar to the Project Plan, with the whole group coming together to do a final careful evaluation of the document before submission. A third-party was also contacted to read over the document and provide their thoughts on how it was written. Several adjustments were made as a result and the final product was deemed to be completed to an acceptable standard by the client.

7.4 Complete Robot Demonstration

Due Date: 27/10/2016
Date Delivered: 27/10/2016

The final demonstration of the robot is used to showcase the final product of the project, with the aim being to prove that the robot meets all the criteria outlined in the SFD provided by the client. The development of the software required by the robot to perform the necessary tasks has been the core focus of the project, with all group members contributing. To ensure that the quality of the task was up to the standards of the client, extensive testing was conducted prior to the meeting. This rigorous final testing, coupled with a variety of other tests completed over the course of the project combined to produce a complete and fully operational version of the robot. During the final demonstration the robot performed exceptionally well and was deemed to meet all the criteria presented by the client.

7.5 Final Report

Due Date: 28/10/2016
Date Delivered: 28/10/2016

The last major document to be handed in is the Final Report itself. The process used to ensure that this report was completed to a high level were comparable to those used for the Project Plan and SDTP. Upon completion of the individual sections by different group members, the entire group came together to evaluate and finalise the document prior to submission. A third party was also used to provide an independent review of the report from a different perspective. Similar to the SDTP, this review resulted in multiple changes being made to different sections of the report. All group members are confident that the level of quality of this report is quite reasonable as a great deal of work has gone into ensuring that it covers the necessary points and provides an adequate explanation of the important aspects of the project.

8. Lessons Learned

The overall difficulty of the project allowed the group to learn a great deal about effectively managing and executing a project for a client. Some of these lessons were:

The importance of design documentation for allowing a team of people to be briefed on what the requirements for the project were and having a document to be able to reference back to. This project was made much easier because of the clarity of the original SFD and the clearly defined requirements the client expected of the final product. As this document is so critical, a lack of detail could lead to multiple problems with the robot not operating in a manner that meets the client's standards.

The thorough initial documentation supplemented the communication within the group and allowed the team to discuss solutions to certain requirements. The team was able to reach a mutual understanding of the requirements and this significantly reduced the amount of time wasted when it came to finding solutions.

The team also learnt about the importance of milestones, as the team found that enforcing specific milestones allowed the progress on the robot to be much more consistent. There were periods of time in which no specific due dates were announced for work being completed. This resulted in generally slower and less effective progress when compared to stages of the project that had clearly defined goals and end dates.

These previous points all lead into the main lesson learned, which was the importance of communication within the group. Upon completion of this project, it is very clear to all group members that strong communication is a vital part of a successful project. Ensuring that all team members are on the same page and are aware of their role within the project is extremely important to producing a final product that meets the client's expectations. Due to how this project involved modules of software being developed in parallel by different group members, there were multiple problems when it came to integrating these modules together. These issues could have been mitigated if there was better communication within the group in regards to what the expected inputs and outputs of a function were to be. Coupling this with ensuring that consistency was present between areas of the code that relied on one another is a part of the project that could have been heavily improved.

9. Recommendations

9.1 Outline of Best Practices

The team found that by having a GitHub[5] repository to store all necessary code and documentation for the robot provided many advantages. The main advantage was that if any program was not working as expected a programmer could always refer to the latest version and start again saving lots of time during the coding process. It also allowed the team to have access to backups of all documentation so that if a document was lost it could always be retrieved from GitHub[5].

Throughout the scope of the project monitoring progress and managing time was essential. This allowed the team to recognize unrealistic deadlines and make the necessary changes to these deadlines based on the current progress. For the deadlines of action items that could not be changed, such as client documentation, the Gantt chart created an awareness amongst team members and motivated the team to focus more closely on achieving close deadlines.

As the project progressed over time the team found that new insights led to constant revisions being made to the original project plan. To prevent many changes occurring to the original project plan it is recommended that a large amount of time should be spent on carefully devising and planning the development of the robot. Doing this would give the team a better understanding of what needs to be done and would reduce the amount of time spent on the coding process, giving future teams more time on other milestones throughout the projects timeframe.

9.2 Suggestions for Improvement

Time management throughout the project was a major issue. Many milestones had deadlines that were set to shorter dates than expected. For instance, while implementing Welcome Mode into the robot, many problems were found to occur with the various module functions that were used. This caused more time to be spent on trying to fixing these modules. To prevent these issues from occurring in future programming milestones a test plan was made for each module to set out how a particular module would be implemented and tested. This ensured that each module satisfied the customer’s requirements before proceeding on with the next part of the development phase.

The implementation of the Global Timer was another issue that raised many concerns during the project. In the future, the occurrence of the problem can be eliminated by testing the Global Timer for a small number of operations. For instance, the Global Timer could be tested by trying to move the robot forward for a short period while displaying a string on the LED display at the same time. Then after passing these test cases the timer can either be implemented into the final design or tested for more complex operations such as running multiple modules at the same time (e.g. run the lights, motor, and LED display all at the same time for any operation). If the Global Timer did not pass these test cases and consumed too much time, then a new approach to the problem would need to be considered. During the project this action needed to be taken to ensure that other risks did not arise, focusing more energy on one milestone would reduce the amount of time expected to be spent on other modules.

9.3 Suggestions for Customer

While designing the software modes for the robot there were some misunderstandings between team members as to what the particular behaviors of modes meant. The wording in the software functional description did not contain enough detail to understand exactly what was required from the team in some instances. Instead the client could have saved a lot of time by providing visual representations to show what should happen during each mode.

10. Appendices

Figure 1 – Gantt Chart of Completed Project

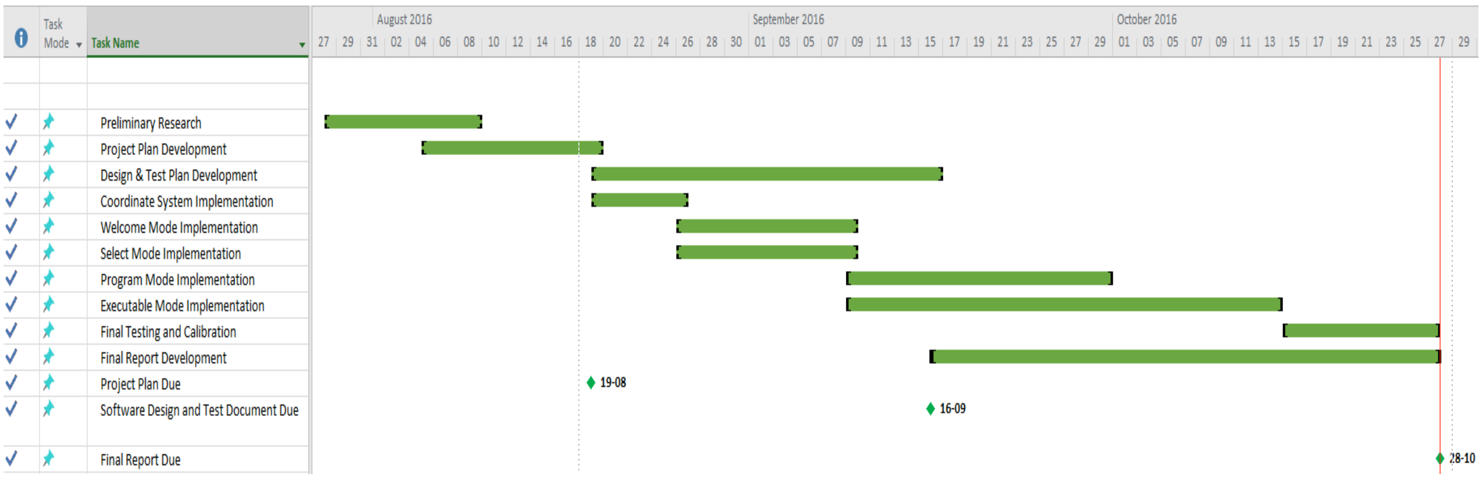

Figure 2 – Full Stepping Sequence [7]

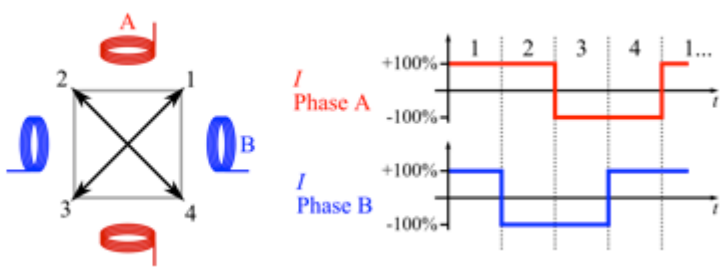

Figure 3 – Sinusoidal Stepping Sequence [7]