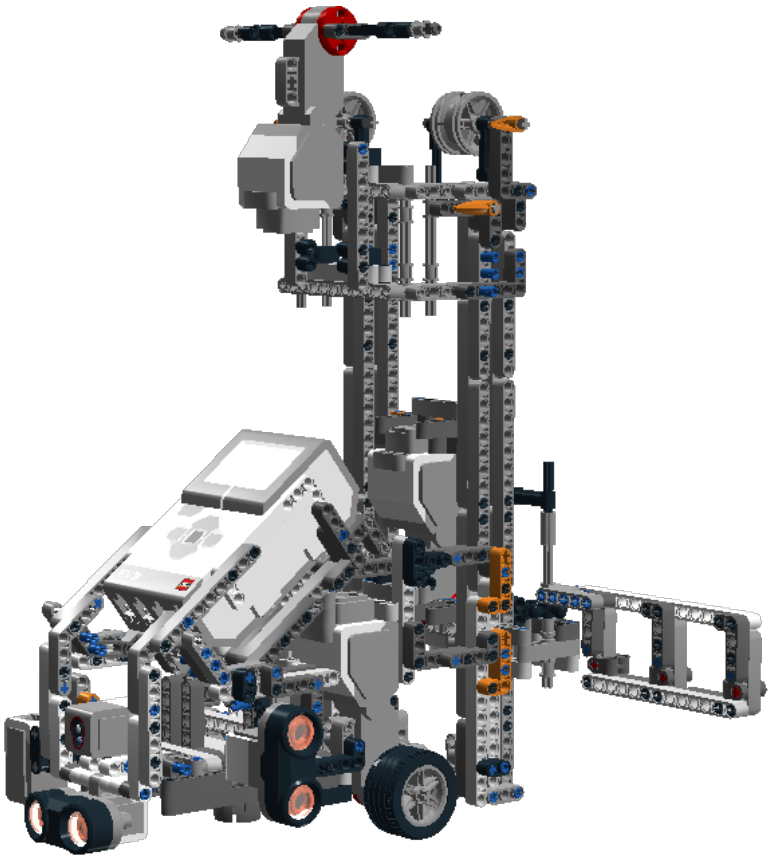

FINAL REPORT

ECSE 211 - TEAM 04

2nd December 2016

Professor Lowther & Professor Giannacopoulos

Kareem Halabi - Project Manager
Jake Zhu - Documentation Manager
Tristan Saumure Toupin - Software Manager
Benjamin Willms - Hardware Manager
Mamoun Benchekroun - Testing Manager
Alexis Gigure Joanette - Software Member
Quentin Norris - Hardware Member
Richie Piyasirisilp - Testing Member

Contents

1 Introduction

1.1 What was the main reason(s) for doing this project?

The purpose of this project was to introduce us to the design process and understand its importance by completing a complex design problem. From start to finish, all eight of us had hands on experience in interpreting and specifying requirements, researching and prototyping various solutions to address the requirements, testing the final design as well as documenting the entire process. This all had to be done with various hardware and time constraints while dealing with the management and communication complications of working in such a large group.

1.2 What was the project intended to achieve?

The technical objective of this project was to design an autonomous robot that is capable of navigating an area, avoiding obstacles, finding blue blocks and stacking them in a designated zone. More specifics about this objective can be found in the *Requirements Document* as well as the *Constraints Document*. This large task had to be broken down into distinct parts that could be assessed, implemented and tested separately. Each of these sub-tasks influenced the design of our hardware and software. Since many sub-tasks were dependent on the previous, once each component was validated, an integration test up until that task could be performed and if validated, the component was added to the final design.

2 Team organization - the start up of the project

2.1 How were tasks allocated?

Tasks were allocated based off of each person's interest, capabilities and workload capacity at the time. These are further described in the *Capabilities Document*. Instead of splitting tasks between 8 people, the project manager assigned them to sub groups of 2-3 people making the task breakdown easier as well as effectively alleviating the workload per person and allowing sub group members to bounce ideas off each other and work together towards a solution. These tasks were distributed by the Project Manager on a weekly basis as to not overwhelm team members if all tasks were allocated at once.

2.2 How was the initial Gantt chart designed?

The Gantt chart was based off of a sample provided by the client at the beginning of the project and adapted to our particular design problem and team structure. Often times, future tasks were heavily generalized (Such as software development) and were updated when the specifics became more evident later on in the design process. All the tasks in the Gantt chart are colour coded (green for documentation, cyan for software, red for hardware, yellow for testing and the default for general tasks) and have resources allocated with the unit of 10% equaling 1 hour. Several tasks (such as hardware and software design) were comprised of collapsible sub-tasks to aid in a broader visual overview of the Gantt Chart. The Gantt chart was also version controlled through GitHub (https://github.com/kareemhalabi/DPM-TEAM04-Gantt) to allow us or the client to monitor changes made to it.

2.3 What information was used to estimate the initial task breakdown?

The initial task breakdown was estimated to match the major deliverables throughout the project which included weekly meetings, the beta demo and final competition as described in *ECSE211-Meetings-Fall 2016*, *Beta demonstration specifications* and the *Project Description* documents respectively. We also consulted with senior students who have previously completed the course for advice on how much time to allocate to various tasks.

2.4 Were any guidelines followed in developing the first version of the chart?

Yes, as specified in section 2.2, the Gantt chart was based off of a sample provided by the client.

3 Issues encountered in the progress of the project

3.1 Were all the dependencies correctly identified at the start of the project?

As specified in section 2.2 many of the tasks were initially generalized because the sub dependencies were unknown. Although many dependencies were correctly identified from the start of the project, we underestimated how dependent the software was on the hardware design. Essentially members of the software team had slack time (with respect to software development) towards the beginning of the project since many of their ideas depended on the final hardware design. They were able to test some localization routines on a prototype test robot however, the code and tests needed to be redone for the final design.

3.2 What dependencies contributed to the critical path of the project?

Most of the critical path was comprised of the construction of the robot, software development tasks and integration testing. Some early documentation tasks, such as the four initial documents were also part of this path.

3.3 What initial ideas turned out either not to work or be based on wrong assumptions?

We were originally debating whether to integrate one or two bricks into our design as initial testing showed that it was reasonable to utilize either case. However when tested on WiFi, the instantiation of remote resources was drastically slower and two bricks would have made the solution more complicated. We also made prototypes of various capture mechanisms that turned out to be unsuccessful: the gravity-fed collector/grabber hybrid had issues with the release mechanism while the platform collector failed due to the fact that no Lego piece was thin enough to scoop up a block. With respect to the chassis, we tried various designs such as two wheels and a caster gear, two wheels and a ball bearing both of which were not effective at crossing discontinuities. Two of our localization strategies: corner bumper and rising/falling edge localization failed due to their slow speed and low accuracy. Towards the end of the project we had issues with Odometry correction that led us to disable it during three of our trials on the final demo day.

3.4 What other issues/factors had an impact on the project

The largest factor having an impact on the project was the availability of resources throughout the semester. Planning meetings was very difficult due to everyone having a different schedule and several members frequently had to prioritize other work throughout the semester.

3.5 How did these affect the project progress?

The debate with one vs two bricks had very little effect since we were prototyping capture designs to work in both cases. However, the chassis issues did impede our progress slightly since it took longer than expected to arrive at a solution that could cross discontinuities and we couldn't test very much without a finalized chassis in place. Thankfully the localization routines were tested on a separate robot and had no effect on our progress. Due to the large number of people on the team, when a member had to shift their focus towards another class, other members were willing to step in to help fulfill the tasks.

3.6 In particular, did the project run to the plan you had initially created?

For the most part, yes. Many of the general tasks such as hardware development, software development and testing were completed in the order we originally planned however many changes to the Gantt chart were made throughout the semester to reflect additional tasks that were not evident from the start. In addition, some tasks were extended or moved around which caused some delays.

4 Budget

4.1 What constraints did the budget place on your team?

As described in the *Constraints Document*, each member was limited to an average of 9 hours per week (not necessarily exactly 9 hours a week). This allowed sub teams to use a large portion of their hours during heavier task loads. Early on, many tasks were hardware related and from the middle towards the end were software and testing heavy.

4.2 How did initial planning for available resources and budget spending affect the development of the timeline?

Based on the interests and skills described in the *Capabilities Document*, each team member was delegated to a sub-team focused on either Hardware, Software, Testing or Documentation. As mentioned in section 4.1, workload per member was not evenly distributed throughout the project, therefore when planning the timeline, it had to be taken into account that members should work more when their sub-team's core tasks needed to be completed and less during weeks when other sub-teams were completing core tasks.

4.3 Did you allocate resources to all the project tasks, i.e. all the way to November 29, at the start of the project and use this to estimate the budget? If not, explain why not.

We originally planned to allocate resources and tasks on a weekly basis until the end of the project however the client requested that this be done for all weeks at once. Therefore, hours were roughly estimated towards the end of the project and were updated with actual numbers from the budget on a weekly basis. This turned out to be a better idea as it prevented us from over allocating hours early on which would have taken hours away from more critical tasks in the later stages of the project.

4.4 What would you have spent if there had been no limits on the budget and when in the process would extra budget have been useful?

Due to the lengthy hardware and software design, we did not have as much time available for integration testing towards the end of the project. In addition it would have been beneficial if additional hours were spent planning the software architecture before writing any code. Additionally resources spent on more Research and Development would have been beneficial to help us better understand the constraints and best use cases of our available tools. Finally in terms of hardware, it would have been beneficial to have an additional kit so that we could construct two copies of the robot allowing multiple teams to work on a robot simultaneously.

4.5 Where were you weak in resources and what would you have done to resolve this issue if you had fewer budgetary constraints? At what point in the project could these extra resources have been brought in?

As mentioned in section 4.4 we could have used additional resources on testing, software planning and research and development. The R&D resources could have been brought in at the beginning, software planning towards the middle and testing resources towards the end.

5 How the process contributed to the success (or failure) of the project

5.1 Was the process useful in achieving the goals?

Certainly, having a formal plan and process to follow allowed us to be better organized, accountable and more aware of any issues early on. Our design process was also quite iterative in the sense that we would come up with an original hardware/software design, test it and then revise it. This was more beneficial than spending a lot of time planning and implementing a single version since many issues don't surface until the solution is implemented at which point they would have been too late to rectify.

5.2 How would you modify the process to increase your probability of success?

As mentioned in section 4.4 if we had an additional kit to construct a copy of the robot, we would be able to work on more tasks in parallel. Hardware could be making improvements on one copy whereas software and testing could be working with a stable copy. Once an improvement was validated by hardware it could be integrated into the other copy. Finally as soon as hardware iterations were complete, both testing and software would have a robot to work on, allowing development and testing to happen at the same time.

5.3 Which parts of the process were the most difficult to implement and why?

By far planning was the most difficult part. For many of us, this was our first design project so we had absolutely no idea exactly what tasks need to be done and how long they should take. Thus many of the estimates of tasks and hours in the initial Gantt chart could not be based off of past experience.

5.4 How much time was devoted to testing?

Approximately 60 to 70 hours were spent on testing.

5.5 Was this at the subcomponent level or did you leave it all to the end?

We ran a combination of component testing and integration testing throughout this project. Each time a component was complete, it was tested individually and usually an integration test was run with other completed components. Towards the end of the project we ran several full integration tests prior to the final demo.

5.6 Were the tests you designed sufficient?

Our tests were designed to validate requirements in the general cases but did not focus as much on edge cases. This was primarily due to time constraints and accepting the fact that edge cases would be rare and not worth investing the time to test and repair.

5.7 How much time did you estimate full prototype (i.e. integration) testing would take?

Integration testing for the beta demo and full demo was originally budgeted for 26 hours.

5.8 How much time did it actually take? If there was a difference, why?

A total of 29 hours were actually spent on beta demo and full demo integration testing. This is primarily because more team members were available to assist for final integration testing.

5.9 How would you change your test design process to make it more effective?

As mentioned in section 5.2, having an additional robot would allow us to run more tests in parallel with the software development, detecting issues early on. In addition more edge case tests could be run in this scenario.

5.10 What was the impact of the beta demo on your design process?

The beta demo was a great milestone for ensuring that we were on track to completing the final design. It allowed us to demonstrate our robot to the client while receiving feedback and making changes based on the results.

6 The success of the Design (Robot) in meeting the original specifications and the performance requirements

6.1 What is your impression of how the robot performed?

Relatively speaking, our robot performed phenomenally. In three out of the four runs, our robot was able to receive the WiFi parameters, localize, navigate to the appropriate zone, avoid obstacles, identify and capture blocks and return them. Unfortunately the blocks may not have been entirely in the zone and we were unable to stack more than one. In addition during one of the runs, the robot completely stopped for no apparent reason which we suspect was due to a firmware glitch as we re-ran the test again with the same parameters and the robot performed as usual.

6.2 Did the robot perform as you expected (i.e. if you wrote down what you thought it would do before the demonstration), did it meet or exceed these expectations?

In terms of reliability we though the robot performed better than expected, only failing catastrophically during one run, however due to the large distances traveled on the final board a larger than expected odometry error caused the blocks to be dropped off not entirely in the dedicated zone.

6.3 If the robot failed (i.e. did not meet all the performance requirements), why did it fail?

The only evident failures were the lack of odometry accuracy and ability to return to the starting corner before the time window had elapsed. The lack of odometry correction was due to an inexplicable, last minute bug that cause the motors to suddenly speed up during the correction procedure, thus we had to disable it for the final competition. We chose not to return to the starting corner since we had not yet implemented avoidance of the opposing team's zone and we were afraid that we may forfeit all our points by entering this zone on our way back to the starting corner.

6.4 Can you point to the sections of the documents that describe the decisions that led to the failure (provide the references to those decisions)?

The odometry correction failure was likely due to the fact that an Odometry Correction component test was not planned out in the Gantt chart and thus was only discovered during final integration tests. The red zone avoidance was also not implemented because other higher priority software components took longer than expected to complete.

7 Conclusions

7.1 What did you learn from this course?

This course has shown us the importance of the design process and how it is useful to transform a set of requirements and constraints into a physical solution. In addition, it has taught us about how crucial effective communication is between members as well as how important time management must be. As our TA Dirk famously told us *"Better is the enemy of done"*. We had to recognize when to stop even if further improvements could be made.

7.2 Explain why a clear, effective and controlled process is necessary when working in a team and what it helped you achieve.

A controlled process allows us all to stay organized and on task. We were able to complete a very complicated and daunting task by decomposing it into sub-tasks doing adequate research to solve them and then integrate the solutions together. Without this process in place, many miscommunications and failures would have occurred.

7.3 Is any of it applicable to other courses you might take?

Yes, not only to other courses but for the rest of our lives working in the industry. Organization and communication with a daunting project such as this is instrumental to the success of any group.

7.4 If so, what and why? (name the courses)

Courses in our curriculum where these processes are a necessity are ECSE 323, ECSE 421, ECSE 425, FACC 400, and ECSE Design projects 1 and 2. These courses require the completion of large projects and collaboration between multiple individuals, similar to that of this class.

7.5 What would you change in what you did if you were doing it over? (important!)

Given the opportunity to redo the project we would be sure to schedule more frequent meetings and complete a more detailed plan earlier. Our first week on the job was in disarray due to an incomplete plan and lack of communication between team members. With each team member already tasked that week with the completion of lab 5, as well as most being caught in the middle of midterm season, this was only expected. For some of us, this was the first time we were working with such a large group of people and we all didn't understand the true value of precise communication. There were many miscommunications among sub teams for that week which resulted in a lot of wasted time. By the second week, we ironed out many of the issues and confusion as well as had a more concrete plan for the rest of the semester. If we had we sorted this out immediately, we would've saved a lot of time. This time saved could have been used to further improve parts of the project which caused us issues during the final demo.

8 Signatures

The undersigned members of Team 4 agree that the contents of both this report and the information handed in on memory keys, provide an accurate representation of the work done in this course and the contributions of each team member.

Kareem Halabi

Jake Zhu

Mamoun Benchekroun

Quentin Norris

Tristan Toupin

Alexis Giguere-Joannette

Benjamin Willms

Richie Piyasirisilp