1. Introduction

Today, autonomous machines are strived for with the belief that they reduce accidents and enable larger profits for corporations with the lower price in comparison to that of human labour. As a result, a push for autonomous machines to replace the task of human labourers is prominent in many fields which range from the automobile industry to manufacturing industries to research fields. [1]

It is with a similar belief that DeteX © was created. Inspired by a nuclear power plant's desire to frequently check for its inventory of used heavy water, which is kept in barrels in a protected area, DeteX © was created. DeteX ©, as presented is a scale-down, proof-of-concept prototype. DeteX © is a mobile platform that can travel along a row of barrels of various types and identify whether each barrel is full, half-full or empty along with the type of barrel (tall or short) and the location of the barrel relative to the "Start Line" while avoiding any obstacles in its path. Although the proposed design is such that all of the design parameters are met as outlined in Section 2, DeteX © was designed with the following criteria in mind:

- Completion of Tasks: The machine needs to perform the tasks required and avoid disqualification. This is defined using a nominal "Yes-No" scale to indicate whether the proposed design completes the function required.
- Ease of Construction: The difficulty in which it takes to construct the machine. Defined using an arbitrary ordinal scale (rating of 1 for most difficult to construct, to 10 for extremely easy to construct). Factors that affect the ease of construction include irregular shapes and the need for precise dimensions.
- Flexibility in Construction: The amount of variance allowable in the design dimensions but still ensuring that the task required is complete. This is defined using an ordinal scale (rating of 1 for least flexible and 10 for really flexible).
- Cost: The price of the final design, with a lower cost being preferred. Cost is defined using a Ratio Scale and uses the Canadian Currency values with USD to CDN conversion rate on January 26, 2016 ($1CDN is $0.70USD).
- Weight: The weight of the robot defined using a ratio scale using the SI units of kilograms.
- Volume: The amount of space the robot takes up both during operation and transportation.
- Time: Time of operation as defined using a ratio scale measured in seconds.
- Aesthetic Appeal (Goal): The appearance of the machine. Aesthetic appeal is defined using an ordinal scale of average member rating upon each member rates the appearance

of the design from 1 being worst looking to 10 being best looking and averaging the ratings.

- Extra Design Features (Goal): The ability for the design to achieve extra design features

Of three possible conceptual designs pitched, the model of DeteX © presented in the subsequent pages best fulfilled the acceptance criteria presented above. Particularly, the Ease of Construction and Flexibility in Construction is what significantly set the selected design apart from the other conceptual designs presented.

The subsequent sections present the complete Design Parameters of the project; further motivation for the project; a complete look at the decision making process; a breakdown of each subsystem with a detailed analysis of how to subsystems were integrated; the entire project schedule; a complete look at the budget and instructions on the machines use.

2. Project Concept and Design Parameters

DeteX © is motivated by a nuclear power plant's need to frequently check for its inventory of used heavy water, which is kept in barrels in a protected area. DeteX ©, as presented in this report is a scale-down, proof-of-concept prototype of a mobile platform that can travel along a row of barrels of various types and identify whether each barrel is full, half-full or empty along with the type of barrel (tall or short) and the location of the barrel relative to the Start Line while avoiding any obstacles in its path. The number of barrels, which is unspecified a priori but is at least 3 and not more than 7, are placed in a straight line must also be identified and presented upon the completion of the run. Detailed design specifications are presented below:

- Each barrel to be detected is a tapered cylindrical container in black colour, with a diameter of $19.5^{\pm 0.5}$ cm at the bottom and $23.0^{\pm 0.5}$ cm at the top.
- Each barrel is equip with a liquid level indicator mounted on its external surface which is emulated by a vertical line made of white duct tape with a width of $7.0^{\pm 0.5}$ cm stretched up from bottom of the barrel.
- Each barrel is one of two types, with the tall barrel having a height of $35.5^{\pm 0.5}$ cm, and the short barrel having a height of$25.5^{\pm 0.5}$ cm. The tall barrel weighs $445^{\pm 5}$ g while the short barrel weights $320^{\pm 5}$ g.
- Barrels are positioned on a line with a ± 1 cm variance.
- Minimum distance between the centerlines of two adjacent barrels is 45 cm.
- There is one round column in black colour with a diameter of $9.0^{\pm 0.5}$ and indefinite height positioned somewhere, unknown a priori, between the barrels along the row. The distance between the centerlines of the column and the barrel is not less than 35 cm.
- Each barrel is oriented such that its level indicator is either at the closest distance with the machine when it passed by, or $180^{\pm 10}$ degrees on the opposite side.
- Each barrel is considered as full when the length of the indicator is more than two thirds of the barrel's height, and half-full when it is between one- and two-thirds of the barrel's height.
- The Start Line is specified no closer than 20 cm, but not farther than 40 cm from the centerline of the first barrel, and the machine must begin its operation from a location such that its front end is at or behind the Start Line.
- The machine can only travel on one side of the barrels during operation.
- Parts of the machine can be in contact with the barrels or column, but cannot move shake, mark, or scratch the barrel or column. Inspection methods are up to the design, but must not require moving or relocating any barrel or column at any time. Each barrel is

considered “detected” only if the machine clearly signals its detection (e.g. light, sound, etc.)

- The machine shall begin operation by hitting <*start*> on the keypad, perform the inspection autonomously until it reaches the seventh barrel or travels 400 cm from the Start Line and then return and stop at the Start Line. Once returned to the Start Line, the machine should be in standby mode displaying a completion or termination message on the LCD and be ready to communicate with the operator. The machine is considered to have “returned to” the Start Line if, at the end of operation, the machine stops where its entire body is behind the Start Line.
- The entire operation shall take no longer than 3 minutes. The operation time is the duration between when the <start> button on the keypad is pressed and when the machine stops and displays a complete/termination message on the LCD. The recorded operation time is considered “correct” if it equals the time measured by the referee $\pm 5\%$.
- The information to be retrieved from the machine after each operation shall include: operation time, total number of barrels for each type, the type of each barrel detected its location relative to the Start Line and the level of liquid in it.
    - The location of each barrel is measured “correctly” if the displayed distance with reference to the Start Line is within ± 10 cm of the real distance (to the barrel centerline)
- The machine must be portable with no need for installations in the field. This is achieved by ensuring that the prototype fits completely within a $55 \times 55 \times 55$ cm³ envelope at all operation times, and weighs no more than 10 kg.
- The machine must have an easily accessible emergency off switch that stops all the mechanical moving parts immediately.
- The machine must use an on-board power supply and be fully autonomous with no interaction with an external PC or remote control during operation.
- The total cost of the machine shall not exceed $\$230$CAD.
- The machine-user interface for both the operation and information retrieval shall be self-explanatory and provide easy navigation for users of various skill levels.
- Each run is “qualified” for scoring if the machine detects at least 3 barrels and attempts to measure their liquid level, does no move shake, mark, scratch the barrels or column, and stops and displays the completion/termination message at the end of its operation and prompts for the inspection information.
- The machine must not pose any hazards to the operator, and shall not be perceived as hazardous (e.g. too much vibration or noise or frequent spike during the operation).

DeteX © aims to succeed in achieving all of the objectives laid out by the project stakeholders as listed above, along with achieving extra design features for improved usability. The added design features DeteX © implements include Real-time Date/Time Display (the date and time of each inspection are displayed on the LCD in standby mode), Permanent Logs (the machine stores log information in permanent (EEPROM) memory and PC Interface (the operation information can be readily downloaded on a PC). In achieving these added design features, DeteX © not only performs the required tasks, but is able to keep track of previous recordings which is extremely important in industry due to various liability concerns.

DeteX © was designed with considerations regarding the preferences listed in Table 2.1 below. Before selecting the final design, many alternative designs were considered. An analytic hierarchy process (AHP) was implemented in selecting the best design when comparing the possible alternative designs with respect to the following objectives, weighted with their relative importance. The model of DeteX © presented is the optimal model in consideration of the preferences listed below.

Preference	Relative Importance
Completion of Tasks: The machine needs to perform the tasks required and avoid disqualification. This is defined using a nominal “Yes-No” scale to indicate whether the proposed design completes the function required. If it is a “No,” the design proposal shall not be accepted.	9
Ease of Construction: The difficulty in which it takes to construct the machine. Defined using an arbitrary ordinal scale (rating of 1 for most difficult to construct, to 10 for extremely easy to construct). Factors that affect the easy of construction include irregular shapes and the need for precise dimensions.	8
Flexibility in Construction: The amount of variance allowable in the design dimensions but still ensuring that the task required is complete. These design changes can be made during construction and building if need be and is extremely valuable as project plans are never perfect. This is defined using an ordinal scale (rating of 1 for least flexible and 10 for really flexible).	7
Cost: The price of the final design, with a lower cost being preferred. Since a lot of the cost is associated with the selection of motors and sensors, cost will be a key component in parts selection in the Subsystem Design Considerations. Cost is defined using a Ratio Scale and uses the Canadian Currency values with USD to CDN conversion rate on January 26, 2016 ($1CDN is $0.70USD).	6
Weight: The weight of the robot. It is noted that it must be less than 10kg to adhere to the design constraints. Once the constraints are met, a lighter robot is preferred to aid in transportability. Weight is defined using a ratio scale using the SI units of kilograms.	5

Volume: The amount of space the robot takes up both during operation and transportation. The machine must fit within a 55x55x55cm envelope at all operation times and a lower volume is preferred. During non-operational times, a lower volume is preferred to increase the ease of operation.	4
Time: Time of operation as defined using a ratio scale measured in seconds. The tasks need to be completed in a 180s window with a lower time being preferred.	3
Aesthetic Appeal: The appearance of the machine. Aesthetic appeal is defined using an ordinal scale of average member rating upon each member rates the appearance of the design from 1 being worst looking to 10 being best looking. Considerations made when deciding Aesthetic Appeal include not having anything sticking out, symmetry and the Keypad and LCD Screen being in an easily accessible place.	2
Extra Design Features: The ability for the design to achieve extra design features particular those aimed at above.	1

Table 2.1: Relative Preferences used in selecting the final design of DeteX ©

3. Perspective

Autonomy and Feedback Systems

Autonomous machines are ones that can act on their own, independent of a controller or any hard coded instructions. [1] The basic ideas behind autonomous machines are to program the machine in such a way that it can sense and response to certain external stimuli. [2] For our machine, it is able to interact with external stimuli by using ultrasonic sensor, infrared sensors and optical encoders. It is programmed in such a way that the ultrasonic sensors are able to detect objects. Detection of an object by the ultrasonic sensor, depending on what sensor detected an object triggers a different response. The best example of this is the highest ultrasonic sensor, located approximately 40cm from the ground as seen in Figure 3.1, which triggers the arm motor to retract the arm. Similarly, the optical encoders, as displayed in Figure 3.2, measure the distance each wheel travels. Noticing a difference between the amount that each wheel travels triggers a response from the machine to slow one of the wheels down and allow for self-correction. It is with this mechanism that the machine is able to keep itself on course and travel in a straight line.

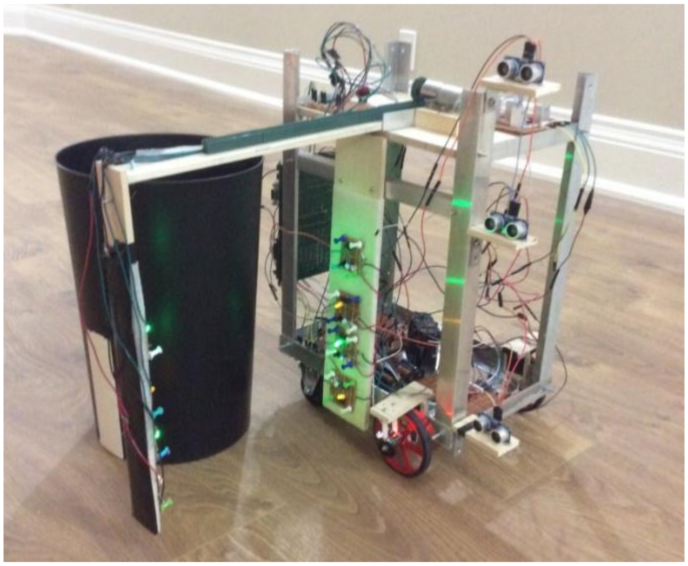


Figure 3.1: An overview of our autonomous machine.

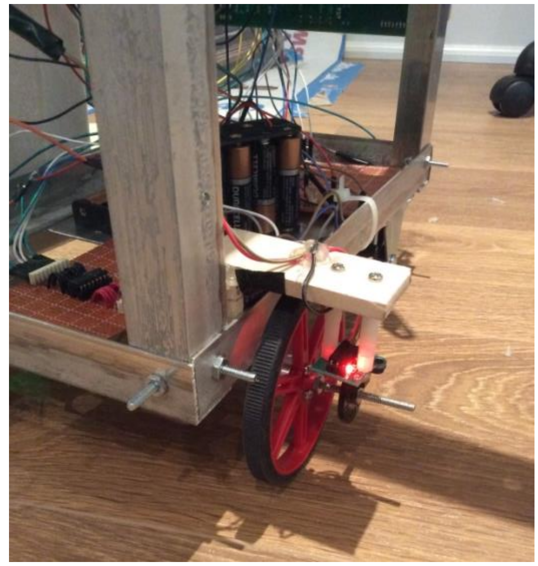


Figure 3.2: Optical Encoders

Autonomous machines have been dreamed about for little under a century. Driverless cars and taxis have been improving the lives of millions in science fiction novels since as early as 1935. [3] In the pre-computer days of the 1930s, autonomous machines, such as autonomous cars were nothing more than the stuff of science fiction. However with the invention of the digital computer, the dreams of autonomous machines suddenly seemed attainable. By 1971 semi-autonomous space probes were landing on other planets. [4] Today, autonomous machines which range from underwater vehicles to driverless cars are at large. Autonomous machines are strived for with the belief that they reduce accidents and enable larger profits for corporations with the lower price in comparison to that of human labour.

It is with a similar belief of the benefits of autonomous machines that a nuclear power plant is looking for an autonomous machine to check its inventory of used heavy water. Having an autonomous machine would not only save money in the long run by reducing the need of labour to frequently do so, but it would eliminate the need of workers going into the protective area where the barrels of heavy water are stored. Although heavy water is not necessarily toxic and harmful, the protected area contains various toxic chemicals and wastes that are toxic to human labourers. Other benefits of machines over human labour is that machines are able to store more information more accurately as they do not depend on human readings which are prone to wrong recordings or incorrect readings; they are able to save years' worth of recordings in an expanding database (unlike notebooks which run out and can get lost); they can easily reproduce and backup recordings on other devices by simply coping bits of information; and they have the ability to check the inventory of used heavy water around to clock, possibly eliminating the need for workers to work night shifts and during holidays.

Limitations and Future Prospects of Machine Response

What constitutes an autonomous machine is not only its methods of detection and self-correction using data and readings collected from sensors, but also how quickly and efficiently it is able to generate an appropriate response. Errors in sensor readings, limitations on motor efficiency, processing delays in the microcontroller, including mechanical losses to friction, heat, and sound all impose restrictions that separate the machine from an ideal solution. Since this machine uses DC gearmotors for all actuation, we focus here on motor limitations. Most small DC motors today have a maximum efficiency of 50-60% [1]. A general calculation of motor efficiency is provided [2]:

First, an adjustment is made for motor, to account for load on its shaft.

$$\tau_l \times \left(\frac{v_{nl}}{C}\right) = v_{reduc} \qquad (1)$$

where τ_l is the torque imposed on the motor's shaft (typically 0.05 kg*cm), v_{nl} is the no-load speed of the motor, C is a proportionality constant (the slope of the motor's torque-speed curve), and v_{reduc} is the reduction overall in speed.

Subtracting the speed reduction from the no-load speed, we get the operating speed

$$v_{nl} - v_{reduc} = v_{op} \qquad (2)$$

To find motor current under load, we use

$$I_{load} = \tau_l \times \frac{1}{D} \qquad (3)$$

where D is the proportionality constant relating current to torque load

Now, the total current drawn by the motor is the sum of the no-load current and the current due to the load

$$I_{total} = I_{load} + I_{nl} \qquad (4)$$

Now, we are able to calculate the output power of the motor:

$$P_{output} = \tau_{nl} \times v_{op} \times K \qquad (5)$$

where K is a constant adjustment

The input power is calculated as

$$P_{input} = V_{applied} \times I_{total} \qquad (6)$$

The efficiency is just $\mu = P_{output}/P_{input}$.

The purpose of this calculation is to show that the motor's efficiency plays a critical role in determining factors such as maximum speed, maximum allowable torque, etc. If the motor is truly 50-60% efficient, then much of the applied voltage is being wasted. Unless the efficiency of small-scale motors increases drastically, more power supply than necessary will be needed to drive motors.

4. Decision Making Process

The decision making process, as presented in the original Design Proposal, with slight modifications is presented below.

4.1. Conceptual Designs

Prior to designing the solution to the presented need, the problem was subdivided into various objectives which need to be addressed in achieving a successful solution. Functional decomposition of the task at hand led to the following subtasks:

1. Go in a straight line for a maximum of 4 metres and return to the starting point. Note that returning to the starting point is defined as stopping with the body entirely behind the "Start Line."
2. Detect the location of a barrel and clearly signal its detection (e.g. light, sound, etc.).
3. Determine the type of barrel (tall or short).
4. Determine the location of the barrel from the "Start Line" to the barrel centerline. Note that the acceptable tolerance is ± 10 cm of the real distance.
5. Determine the heavy water level in each barrel (full, half-full or empty).
6. Detect the location of a column.
7. Avoid the column and resume regular operation thereafter.
8. Display the operation time (defined as the duration between when the <*start*> button on the keypad is pressed and when the machine stops and displays a completion/termination message on the LCD), total number of barrels, the type of each detected barrel, its location with reference to the "Start Line," and level of liquid in it.

It is noted that the completion of the listed tasks would satisfactorily address the need at hand. Further design constraints were imposed by the client in order to be applicable to the certain specifications of the particular power plant. These constraints are as follows and needed to be addressed in order to have an acceptable and feasible solution:

1. Completely fit within a 55 x 55 x 55 cm envelope at all operation times.
2. Weight of the machine does not exceed 10 kg.
3. Cost shall not exceed $230CDN.
4. Machine must use its own on-board power supply during the operation.
5. Machine must be fully autonomous with no interaction with an external PC or remote control allowed during the operation. The operation must start by hitting a <*start*> button on the keypad.
6. Must have an emergency STOP button that stops all the mechanical moving parts.
7. Operator must be able to set up the machine for operation and take it away afterward conveniently. No instrumentation is allowed in addition to what is devised within the machine.
8. Parts of the machine cannot move, shake, mark or scratch any of the barrels.
9. Machine-user interface for both the operation and information retrieval shall be self-explanatory and provide easy navigation for users of various skill levels.
10. Must detect at least 3 barrels and attempt to measure their liquid level.
11. Avoid disqualification by not falling over, jamming unpredictably, failing to test at least 3 barrels, moving/shaking/scratching the barrels/column or does not display the termination/completion message on the LCD at the end of operation.
12. Must not pose any hazard to the operator and shall not be perceived as hazardous.

In addition to the basic design objectives and constraints presented above, DeteX © aims to be the best proposed robot by additionally realizing the following Design Features:

1. Elegance and Safety: machine looks elegant, and operates quietly and smoothly with little or no sensible noise or vibration.
2. Real Time Date/Time Logs: date and time of each inspection are displayed on the LCD in standby mode
3. Permanent Logs: machine stores log information in permanent (EEPROM) memory
4. Remote Operation: machine can start and (emergency) stop by a remote controller
5. PC Interface: operation information can be readily downloaded on a PC

The remainder of the section outlines the alternative proposed solutions and the decision making process that ultimately led to the final design. Furthermore, detailed decisions of each subsystem are also addressed.

It should be noted that the final product was only able to achieve the Real Time Date/Time Log Extra Design Feature.

4.1.1. System Design Considerations

The System Design Considerations are those regarding the large scale design features. Detailed Design Considerations were not completed at the time due to the member's inexperience with such projects. As a result, a large scale overview was presented with the detailed selections of all the materials used determined as time went along. As a result, the System Design section has general solutions and design placements but refrains from heavy technical details about each design such as material selection. Each conceptual design is described below with focus on achieving the required tasks.

4.1.1.1. Conceptual Design #1

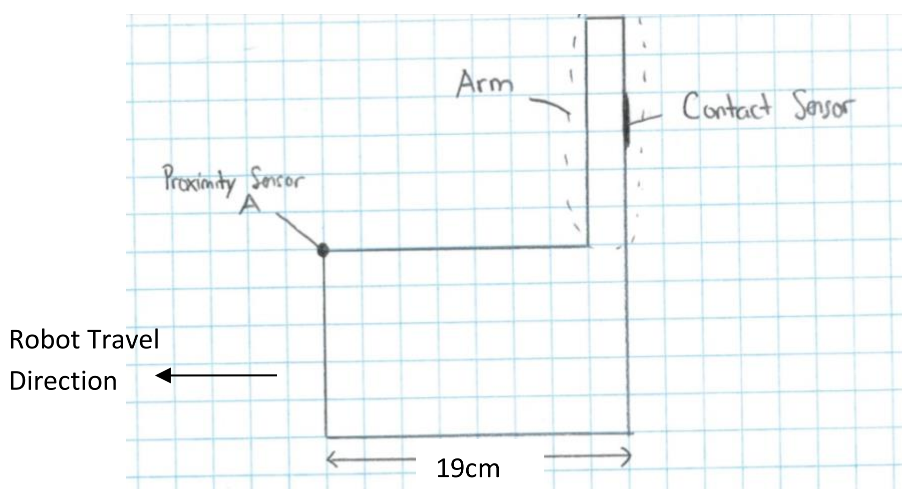


Figure 4.1.1.1.1: Side profile of design

The main design features of this design are the methods in which to detect the location of the column and column avoidance, along with ensuring the body returns to the "Start Line." For the detection of the column, a sensor, indicated by Proximity Sensor A in the figure above, is placed at the front of the machine at a height greater than 35.55cm thereby avoiding the tallest barrel and less than 55cm to fulfill the project constraints. As a result, the proposed height is 40cm. An Ultrasonic Sensor was proposed to be used as the Proximity Sensor due to its ability to detect objects irrespective of their colour. The column avoidance mechanism works by having the Arm retract fully back. As soon as Proximity Sensor A detects the presence of an object, the Arm retracts back. Knowing that the minimum distance between the centerlines of the column and the nearest barrel is 35cm, radius of the column is 4.5cm and maximum radius of the barrel is 11.5cm the machine needs to be less than 19cm wide to ensure that the Arm can retract without hitting or scrapping the barrel.

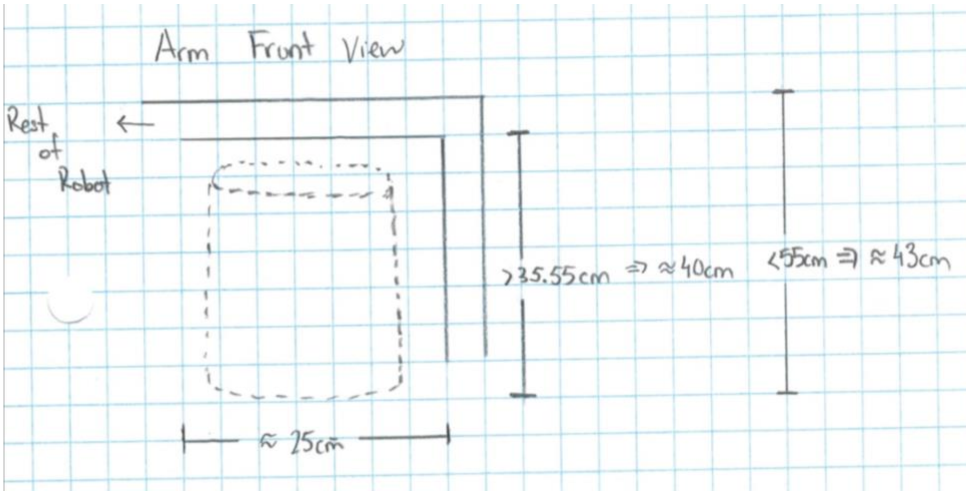


Figure 4.1.1.1.2: Front profile of arm

The proposed dimensions of the Arm are depicted in the figure above. The inner height of the Arm must be greater than 35.55cm to clear the height of the tallest barrel with a proposed height of 40cm thereby provided a factor of safety for all the wiring and sensors. The inner width of the Arm is proposed to be approximately 25cm with the outer width being approximately 30cm. This in turn leaves 25cm for the width of the base thereby fitting within the 55cm width constraint.

Sensor S/T 1	Sensor S/T 2	Result
0	0	Empty
1	0	Half-Full
1	1	Full
0	1	Not-Possible

Note: 0 represents not detected and 1 represents detected.

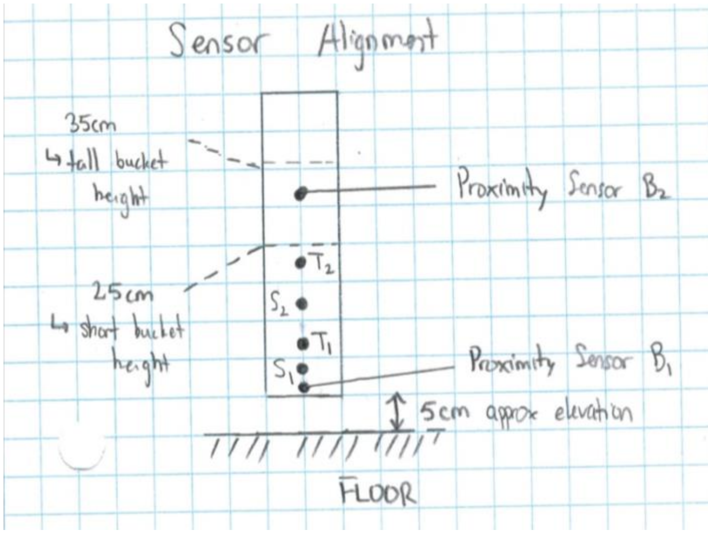


Figure 4.1.1.1.3: Alignment of the sensors on the arm

Table 4.1.1.1.1: Water Level Sensor Logic

The proposed design for the detection of barrel and water level is the sensor alignment as depicted in the figure above. Proximity Sensor B_1 (also an Ultrasonic Sensor as in Proximity Sensor A) is used to detect the presence of the barrels. Due to the fact that the Arm will be approximately 5cm from the ground, the Proximity Sensor can be all the way on the bottom of the Arm thereby detecting the presence of the barrel. A similar sensor, indicated as Proximity Sensor B_2 is used to determine the type of barrel – tall or short. The logic is quiet simple – if Proximity Sensor B_2 detects an object at the same time that Proximity Sensor B_1 detects an object, than the barrel is tall. If Proximity Sensor B_1 detects and object, but B_2 doesn't, then the barrel is short. It should be noted that Proximity Sensor B_2 is placed approximately 30cm from the ground thereby ensuring that the shorter barrel is cleared.

Determining the level of water is achieved by the sensors S_1, S_2, T_1, and T_2 which are meant to be able to detect white. Infrared Sensors are selected for this task. It is noted that sensors S_1 and S_2 are placed at 1/3 and 2/3's the height of the short barrel at 8.33cm and 16.66cm from the ground while sensors T_1 and T_2 are placed at 1/3 and 2/3's the height of the tall barrel at 11.83cm and 23.66cm from the ground. The logic for determining the water level is discussed in the table above. It should also be noted, that Proximity Sensor B_2 is used as a multiplexer to select which sensors are used to determine water level – if B_2 detects an object, the T sensors are used; else, the S sensors are used. Note that the same sensor logic is also used on the body of the robot in case the water level indicator bar is in that orientation. To reduce cost, proximity sensors B_1 and B_2 only need to be present on one side – on the robot or the arm.

Lastly, ensuring that the machine returns to the start line is achieved by retracting the Arm completely inside the robot body and reversing the distance of forward travel, plus a safe factor of safety in case of turns to ensure that the machine's body is entirely beyond the start line. The robot knows to turn back after it travels forward 4m or detects seven barrels. The mechanism for straight travel is discussed in greater detail in subsequent sections and is the same for all of the proposed designs.

4.1.1.2. Conceptual Design #2 – Selected Design

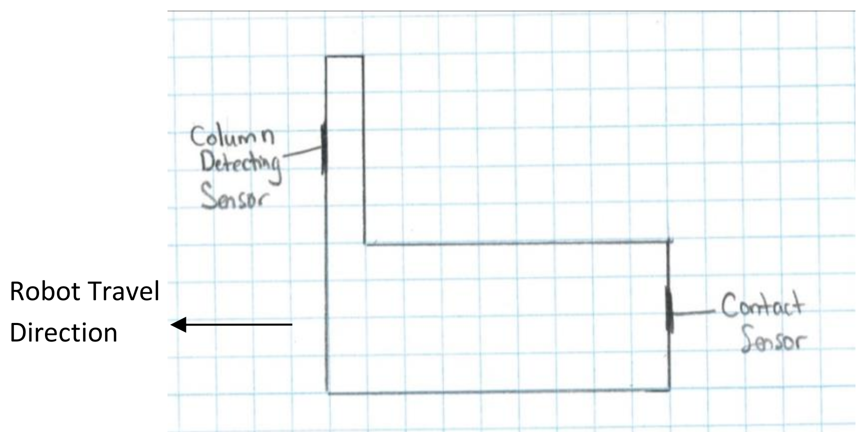


Figure 4.1.1.2.1: Side profile of machine

The second conceptual design was similar to the first design with the change in the position of the Arm. The only change with the second design is that the column detecting sensor is placed on the arm and the column is detected. With this design no restrictions needs to be imposed on the robot body expect the 55 cm length constraint. The dimensions of the Arm and width are the same as the proposed design above.

It should also be noted that both Design 1 and 2 have four-wheel drive with the wheels placed at the front and rear. Wheel placement and selection is discussed further in the Electromechanical Subsystem. The wheels could both be placed directly to the side or under the main body of the machine and will be determined by the Electromechanical member.

4.1.1.3. Conceptual Design #3

The third conceptual design proposes a machine that is similar to the past conceptual designs in the drive mechanism and overall functionality, but differs in the number and location of the sensors. Two proximity sensors are used for column avoidance, where one is mounted in the middle of the horizontal portion of the arm and the second is placed on the top corner of the body closest to the column. By utilizing 2 proximity sensors to detect the presence of a column, the risk of the column being undetected is reduced. While the sensor on the arm is oriented to face the same direction as the direction of travel, the sensor on the machine is placed at an angle such that the column is only in its field of vision in a small duration of time. This allows for measuring changes in at 2 independent points, instead of 1.

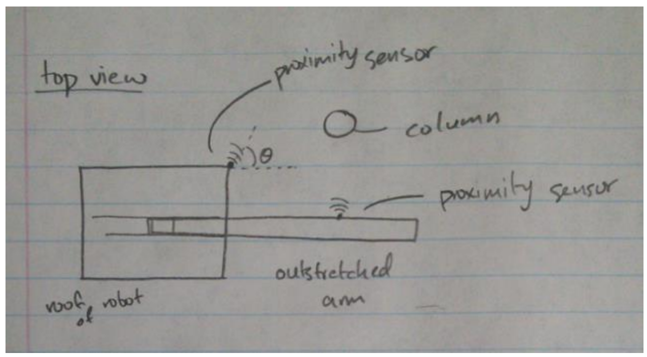


Figure 4.1.1.3.1: Two proximity sensors for column avoidance**..**

In addition, 2 proximity sensors are used to distinguish barrels of different size. These sensors are placed on the side of the body closest to the barrels at the base of the robot with the same relative heights as conceptual design 1. Locating the sensors here ensures that both sensors are equally accurate since the distance separating the robot body and the barrels is equal along the vertical direction, while the arm may create a slight angle when outstretched, making one sensor slightly further away.

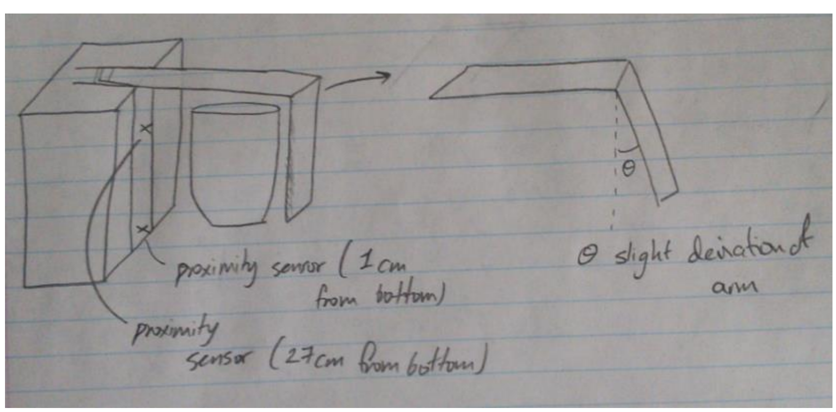


Figure 4.1.1.3.2: Two proximity sensors for barrel differentiation.

As for the measurement of the level of the liquid in each barrel, it is proposed that a continuous stream of sensors, with a total of 10 equally spaced sensors on both the arm and the body, be used. Instead of allowing the results of one sensor to determine which sensors are activated, the level is measured independently, and with greater accuracy. Although this introduces more complexity into the microcontroller subsystem, it aims to minimize the risk that the level may be incorrectly measured.

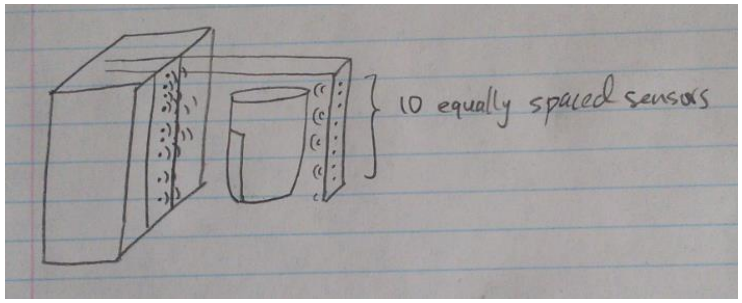


Figure 4.1.1.3.3: Continuous stream of sensors.

Since the widest diameter of both barrels is less than 26cm, it is proposed that the width of the base will be 23 cm, allowing the arm to stretch outward at least 30cm horizontally. This is a sufficient factor of safety to ensure no contact is made between any components of the machine and the barrels. The length of the base of the machine is to be 23cm as well, to allow for symmetric construction.

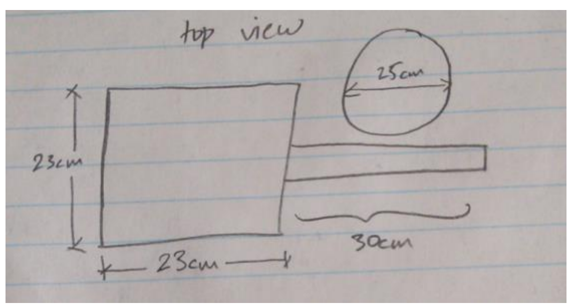


Figure 4.1.1.3.4.: Geometry of the framework and arm

4.2. System Decision Making

The objectives used as a method of comparison in selecting which design is the best, and their Relative Importance (RI) is assigned in the table below.

Objective/Preference	Relative Importance
Completion of Tasks (Objective 1): The machine needs to perform the tasks required and avoid disqualification. This is defined using a nominal "Yes-No" scale to indicate whether the proposed design completes the function required. If it is a "No," the design proposal shall not be accepted.	9
Ease of Construction (Objective 2): The difficulty in which it takes to construct the machine. Defined using an arbitrary ordinal scale (rating of 1 for most difficult to construct, to 10 for extremely easy to construct). Factors that affect the easy of construction include irregular shapes and the need for precise dimensions.	8
Flexibility in Construction (Objective 3): The amount of variance allowable in the design dimensions but still ensuring that the task required is complete. These design changes can be made during construction and building if need be and is extremely valuable as project plans are never perfect. This is defined using an ordinal scale (rating of 1 for least flexible and 10 for really flexible).	7
Cost (Objective 4): The price of the final design, with a lower cost being preferred. Since a lot of the cost is associated with the selection of motors and sensors, cost will be a key component in parts selection in the Subsystem Design Considerations. Cost is defined using a Ratio Scale and uses the Canadian Currency values with USD to CDN conversion rate on January 26, 2016 ($1CDN is $0.70USD).	6
Weight (Objective 5): The weight of the robot. It is noted that it must be less than 10kg to adhere to the design constraints. Once the constraints are met, a lighter robot is preferred to aid in transportability. Weight is defined using a ratio scale using the SI units of kilograms.	5
Volume (Objective 6): The amount of space the robot takes up both during operation and transportation. The machine must fit within a 55x55x55cm envelope at all operation times and a lower volume is preferred. During non-operational times, a lower volume is preferred to increase the ease of operation.	4
Time (Objective 7): Time of operation as defined using a ratio scale measured in seconds. The tasks need to be completed in a 180s window with a lower time being preferred.	3
Aesthetic Appeal (Objective 8): The appearance of the machine. Aesthetic appeal is defined using an ordinal scale of average member rating upon each member rates the appearance of the design from 1 being worst looking to 10 being best looking. Considerations made when deciding Aesthetic Appeal include not having anything sticking out, symmetry and the Keypad and LCD Screen being in an easily accessible place.	2

Extra Design Features (Objective 9): The ability for the design to achieve extra design features particular those aimed at above.	1

Table 4.2.1: Acceptance Criteria: Objectives and Relative Importance

To decide what conceptual design is the best one, each design is compared against the others using an Analytical Hierarchy Process in which the three designs are assessed against each other objective by objective. The Relative Preference of each design is assigned for each objective and ideally the design that is preferred for the most objectives (or the most important objectives) is selected. The performance of each design is assessed and justified in Appendix D – AHP Decision Making Details. After extensive calculations and verification of the decision making consistency, as outlined in Appendix D, it was concluded that Design #2 is the best system design. In the next section, each subsystem is considered separately and specific design decisions and how they were incorporated are presented in the next section.

Intuitively, it makes sense why Conceptual Design #2 would be selected as the design to move forward based on the Acceptance Criteria presented in Table 4.2.1 above. Due to the teams value of Ease of Construction and Flexibility of Construction, it makes sense as for why this design would be moved forward since it did not have any constraints, as opposed to the other conceptual designs presented. For example, Conceptual Design #1 needed to be less than 19cm in length, putting a limit on the size of the robot which was not desirable.

5. Subsystems

From weeks 1 to 8, the project was subdivided in the varying subsystems. Most of weeks 1 to 4 involved project planning and close communication among the team members in completion of the Design Proposal. Simultaneously each team member was responsible for a particular subsystem in which they needed to run various tests in order to make conscious design decisions and start working towards completion of their own subsystem designs. A summary of each member's subsystem and responsibility is listed below:

- Amr Mahmoud: Processing, Control and Communication Subsystem (Microcontroller)
- Hamza Nisar: Structure, Mechanism and Actuation Subsystem (Electromechanical)
- Stefan Momic: Power, Instrumentation and Interfacing Subsystem (Circuits)

It should be noted that during the individual subsystem phase of the project, communication between the team members was not lost and constant discussion was held in order to complement each other's subsystem designs to allow for an easier task of integration. A simple example of this include communication between the electromechanical and circuits member in the kinds of encoders and motors intended to be used in order to ensure that both members obtained the right shafts/mounts and circuit elements respectively. A more detailed analysis of tasks completed by each subsystem from weeks 1 to 8 is listed in the subsections below.

5.1. Processing, Control and Communication Subsystem

The control and communication subsystem deals entirely with programming the microchip to perform the required tasks. Controlling all of the motors and sensors to communicate with each other effectively and smartly requires a very inclusive code that takes everything into account. The robot has to be programmed for all tasks in all of the operating conditions and the result is an extremely long code that can be seen in Appendix X. DeteX © accomplished almost all of the tasks individually but came short when the processes were integrated together and the code was compiled in one big file. The likely cause of failure is that the program got stuck somewhere in the interrupt or randomly interrupted in an unaccounted for manner. This resulted in DeteX © freezing and only inspecting two barrels at the demo without continuing the rest of the inspection. Even though DeteX © failed at the demo, the code is extremely well refined and, with some further development, will work to the fullest capacity, completing all the tasks and even applying one of the extra design features.

1. Travel in a straight line for a maximum of 4m and return to the 'Start Line'
2. Detect the location of a barrel and clearly signal its detection
3. Determine the type of barrel (tall or short)
4. Determine the location of the barrel from the 'Start Line' to the barrel centerline.
5. Determine the level of heavy water in each barrel
6. Detect the location of the column and retracting the arm to avoid contact

Task#1: drive in a straight line for a maximum of 4m and return to the start line

The task of driving in a straight line and returning back to the start line after 4 meters of traveling is accomplished through the use of the optical encoder in combination with interrupts. The optical encoder works through the use of an IR transmitter and receiver. It comes with a circular wheel that has 20 equally separated openings. The optical encoder transmits an IR beam into the receiver and has the signal blocked when the IR beam strikes the solid part of the wheel. This creates a signal of 1's and 0's, 1 when the IR receiver is receiving incoming IR waves and 0 when there is no IR waves being detected. To program for these conditions, two timers, Timer0 and Timer1, are enabled and every time they throw an interrupt, the pins where the encoders are have their values checked. The old value is compared to the new value, and if the two are different a count is increased and tracked to determine how many time each wheel turns. Another way this task could be accomplished is by using the external interrupt or interrupt on change pin that throws an interrupt if a pin changes states from high to low. This method was voted against simply because there is only one interrupt on change pin and it would be hard to keep track of both wheel movements with only one pin. For the task of going straight, the count created above for each of the wheels is compared with every interrupt. If the counts differ (the robot is turning one way or the other), the appropriate motor is stopped in order to allow the other motor to spin (and the robot to turn) until the count is the same. A quick and easy way to check the functionality of this logic is to run the code and stop one wheel. If the other wheel stops moving as well after a little turn, the code is doing its job correctly. Even though this code is logical and should work perfectly in theory, a good electromechanical job is required for the code to be fully effective. A major reason why DeteX © failed to complete the task of going straight is a result of a poorly mounted encoder and imperfect wheel. The encoder was mounted onto a shaft that is supposed to extend from the center of the wheel but the drilled hole is slightly deviated from the center resulting in the shaft and encoder oscillating in an unwanted manner. The encoder moves up and down as well as left and right causing a number of different problems and the wheel ended up cracking/deforming because of the drilled hole. Even with this perfect code, DeteX © deviates from a straight path to the right as it travels. To counter-act this in a purely mechanical way, I recommended taping down/closing some of the openings on the right wheel to make the program think that it is spinning way less than the left one and cause it to turn right to correct the path straight. The way I handled the 4 meter problem is just a matter of simple calculation and implementation. I measured the diameter of the wheel and counted the number of separations on the encoder. Through some simple math shown below, it is possible to calculate the number of turns the wheel has made in a 4 meter run and create a function that checks the count and stops the motors when the robot has traveled 4m. The number of turns required was calculated to be 509 which was rounded up to 510 for safety and since 8 bit numbers can only go up to 255, a second variable (Countr) was required. Countr is set to 1 if the Count variable reaches 255, which then Count is set to 0 and the program constantly checks for when it reaches 255 again so it can stop the motors (4m reached). Logic tables for the above code can be found below. One and Count refer to the variables that store the number of wheel turns on the left and right wheels respectively. The encoder code can be found in the interrupt section and the check distance function.

$$\text{Circumference} = \pi * d = \pi * 5 \text{ cm} = 15.71 \text{ cm}$$

$$\text{Distance traveled/turn} = \text{circumference} / 20 \text{ turns} = 15.71 \text{ cm} / 20 = 0.785 \text{ cm / turn}$$

$$\text{Turns required for 4 m} = 4\text{m}/(0.785\text{cm /turn}) = \mathbf{509\ turns}$$

One - Count	0	1
Z (set if equal)	Check C bit	Go Straight
C (set if there was a carry)	Turn Left	Turn Right

Table 5.1.1: Turning Logic

Task#2: Detect the barrel and clearly signal its detection

Task 2 was accomplished through the use of ultrasonic sensors and two led lights. There are 3 ultrasonic sensors that can be seen clearly mounted to the front of the robot. The top sensor is above the tall barrel in order to detect the column while the other two are used to detect barrels and their type. Ultrasonic sensors were a particularly nasty sensor to work with for the pic since they were meant for use with an Arduino. Even so, they are capable of functioning perfectly with the pic16f877 and that was the task I was burdened with. Even though I started coding on this sensor since week 2, it took a significantly long time (week 9) to get it to work correctly. The ultrasonic sensor works by sending a 10 micro second 5v signal as a trigger (Trig function) to the sensor and waiting for an echo to come back then timing how long the signal is high for (WaitEchoHigh). The longer the signal is, the farther away the object is. The code uses the ultrasonic sensor to detect both barrels and columns when they come less than 10 cm from the sensors. Another big problem with the ultrasonic sensor was that it would sometimes give false readings upon start up and get the robot thinking that it detected a column constantly. To counter this, double checking the readings was implanted in which the trigger was sent again and the echo time was recorded from scratch. CheckUltra is a function that checks, double checks, and even triple checks the ultrasonic sensors readings to ensure the highest level of accuracy. DetectedBarrel3 waits until the ultrasonic sensors at the front no longer read anything so that the barrel is in the perfect position for the IR sensors to read the water level. Upon detection, bit 3 of port B was set to high which turned on the two LEDs that are connected in series. The LED remains on for the duration of the period that the barrel is detected.

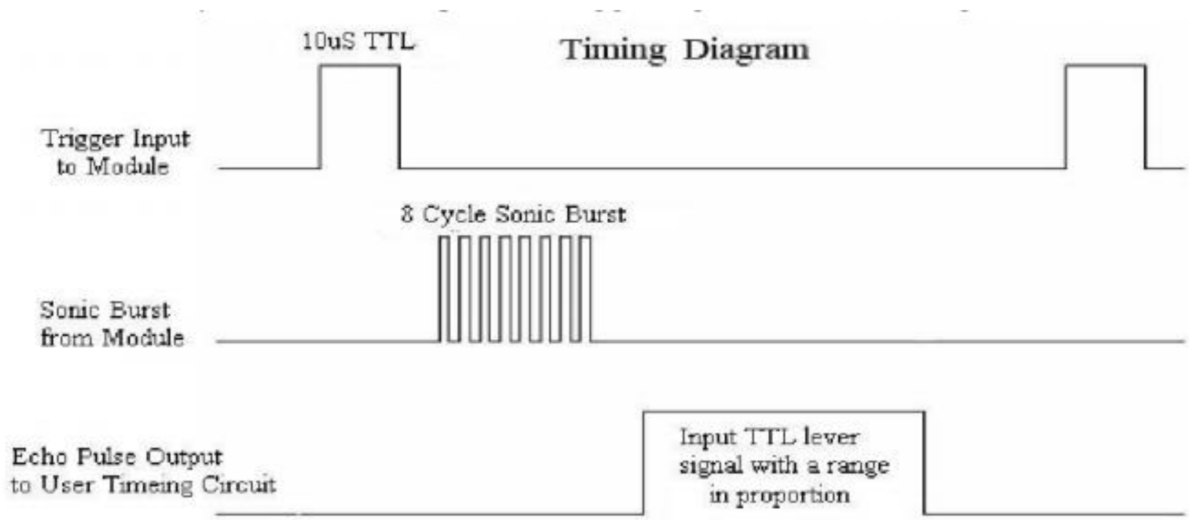


Figure 5.1.1: Ultrasonic sensor timing diagram.

Task#3: Determine the type of barrel (tall or short)

The barrel type was also determined through the use of ultrasonic sensors. The two ultrasonic sensors that are below the one used for detecting columns are used to detect barrels and their type. The middle ultrasonic sensor lines up with the height of the tall barrel while the bottom ultrasonic sensor is positioned below the height of the short barrel. The logic behind the ultrasonic sensors can be seen in tabular form in the table below. In the case that the bottom sensor detects something and the top sensor doesn't detect a column, the middle sensor is checked. If the middle sensor detects something, there is a tall barrel, otherwise the detected object must be a short barrel. This code is fully implemented in the CheckUltra and DetectedBarrel3 functions. Depending on the barrel type, DeteX © is programmed to uses and reads the values of different sets of IR sensors that line up at the appropriate 1/3 and 2/3 of the barrel's height.

$U_3U_2U_1$	Meaning
001	Short barrel detected
011	Tall barrel detected
111	Column detected

Table 5.1.2: Ultrasonic sensor readings, 1 being detected and 0 being not detecting. U3 is the top sensor, U2 is the middle sensor, and U1 is the bottom sensor. There are only 3 possible cases that are used when the ultrasonic sensor detects something which are shown in the table.

Task #4: Determine the location of the barrel from the 'Start Line' to the barrel centerline.

The location of the barrel was a particularly hard task that wasn't fully implemented and functional in DeteX ©. The appropriate way to implement this functionality is simply to read the distance traveled by the wheels through the encoders in cm. This proved to be tougher than expected since, at the byte level, decimal multiplication is difficult. As a result, DeteX © takes a simpler and more accurate measure of the distance of the barrel by turning the encoder values into a Binary Converted Decimal (BCD) and saving it to the EEPROM memory to be accessed and read at the end of inspection. Even though this will require that the user manually do calculations to determine each barrels appropriate location (multiply by 0.785 cm/turn), it is a quick and easy way to avoid a difficult process.

<u>Task #5: Determine the level of heavy water in each barrel</u>

The level of the water in each barrel is implemented through the use of IR sensors. Once the type of barrel is determined by the method stated above, the appropriate IR values are read and added up to determine the level of water in each barrel. The IR sensors work in a similar manner as the encoders. There is a transmitter and a receiver that constantly emit and receive IR waves. When the receiver is receiving IR waves, the voltage is high (usually >1V) and when it is not receiving any waves it is approximately 0. It was found through experimentation that if the IR transmitter transmits waves onto a black surface, they are almost perfectly absorbed and reflect nothing back while when IR waves strike a white surface they are almost fully reflected completely. This information was then used to develop a sensor that can detect the presence of white tape on the side of the barrels with ease. More specific detail about the sensors can be found in instrumentation and interfacing subsystem information as well as the logic table that dictates how the water level of each barrel is read. CheckStatusTall and CheckStatusShort both function to check the water level of the barrel by reading the appropriate values of the IR sensors which will be set to 1 if they detect white and 0 if they detect black. By summing up the total of IR sensors detecting white and recognizing that the tape can only be on one side, it is easy to show that if the value of stat1 (the counter) is 2, then the barrel is full, if the value is 1, the barrel is half full, and if the value is 0, the barrel is empty. The status is then written to the EEPROM to be accessed and read at the end of inspection.

<u>Task #6: Detect the location of the column and retracting the arm to avoid contact</u>

The detection of a column works in the exact same way by which the barrels are detected. Since the top ultrasonic sensor is above the height of the tall barrel, if it detects something it has to be a column by simple deductive logic. The CheckUltra function checks if the top ultrasonic sensor detects something by checking to see if the echo value is less than 10 cm. If it is, the function DodgeColumn is called in which the arm is retracted, DeteX ©moves forward for a second, and the arm is extended again and inspect is called to restart the inspection loop. Dodging the column is a relatively simple task and was immediately implemented as soon as the ultrasonic sensors were fully functioning.

5.2. Structure, Mechanism and Actuation Subsystem

The Structure, Mechanism and Actuation Subsystem was completed by Hamza Nisar, *April 5, 2016*

The structure, mechanism, and actuation subsystem, hereby referred to as the electromechanical subsystem, encompasses the machine's mobile base, physical structure, detection system for the barrels and barrel liquid levels, and a column avoidance mechanism. An extendible arm allows for simultaneous detection of either side of the row of barrels while the machine's travel is confined to a single side of the barrels. The arm is retracted and extended as necessary, avoiding the column in its path while maintaining its original position for barrel inspection.

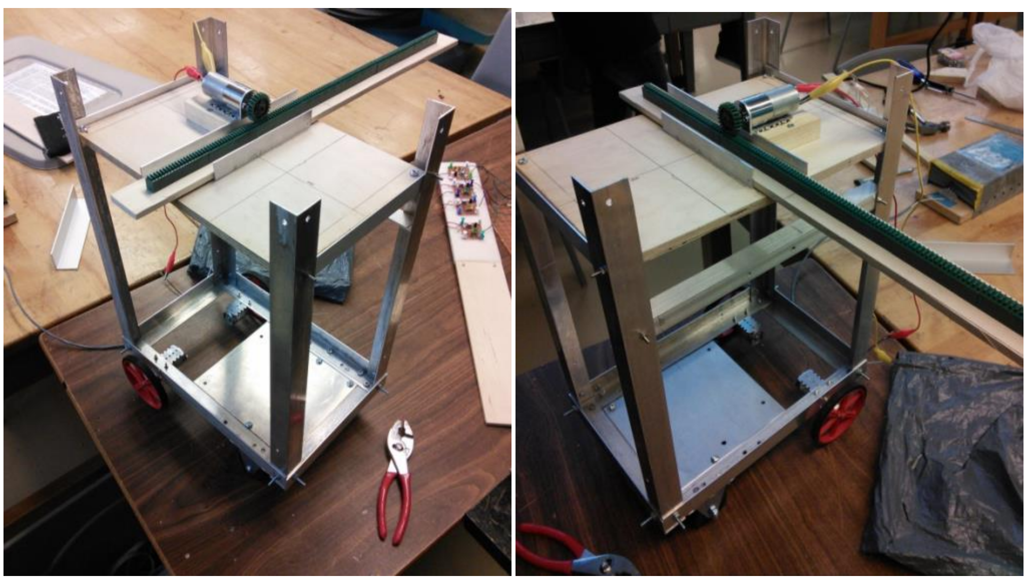


Figure 5.2.1 (a) and (b): The Electromechanical Subsystem, with the absence of the vertical arm, shaft encoders, and sensor balconies.

Indeed, a large variety of viable electromechanical solutions are possible for this task, but the final design aims to minimize the actuation and moving parts within the design constraints. A machine, the majority of whose components are static, allows the designer to obtain a more accurate analysis of its functionality, allows for efficient and rapid construction, integration and debugging, and reduces possible mechanical failures. It is also essential to be aware of resource constraints such as the time allotted, budget, weight, operating space, and the practical experience of the designer and the limitations they impose on the feasibility of the design.

5.2.1. Assessment of the Problem

The final, working solution is one that incorporates all three subsystems to complete each of the desired subtasks. To understand the electromechanical problem, any subtask must be analyzed in regards to the contribution of the physical structure and motional components. As such, a list of the major subtasks are given below, each accompanied by the appropriate electromechanical requirements.

- The machine is autonomous and moves in a straight line, for a maximum distance of 4m, and returns to its starting position, completely past the start line.
    - → **Autonomy:** The structure of the machine is capable of housing all required circuit components and power supplies. There is sufficient space and carrying capacity for the sensors and the microcontroller board to be mounted in pre-specified locations.
    - → **Travel in a Straight Trajectory**: The wheels on the base of the machine are able to carry the load and the machine's self-weight without slipping. In addition, there must be correction mechanisms in place for any deviations from the trajectory. The wheel actuators are able to supply enough torque and rotational speed to keep the total duration of the run and the return trip under 3 minutes.
- The machine is confined to a single side of travel, but it must be able to detect either side of a barrel in case a level indicator appears on the opposite side. Its operation must not be hindered by the column of indefinite height that is placed randomly in the line of barrels.
    - → **Level Inspection:** The machine comprises of a subassembly that allows for simultaneous detection of both sides of a barrel. Such a mechanism is constructed to remain within the space constraints, is able to hold the appropriate sensors needed to detect the liquid level, and the functionality of which is not impeded by the location and placement of the tall column.
- The machine is able to determine the physical size and relative distance of the barrels from the start line without moving, shaking, marking, or scratching the barrels. Since the location of the column is unknown beforehand, it must be detected during the robot's real-time operation so that the machine can respond accordingly.
    - → **Barrel & Column Detection:** A detection system is in place that utilizes appropriate sensors to determine whether the next object is a small or large barrel or a column. If at any point the machine should stop its forward motion, the detection system must account for a stopping delay equal to the distance travelled by the wheel actuators.

In the following section, the electromechanical solution is described in its entirety. Design considerations that are independent of the initial RFP constraints are also mentioned. Details of the costs of the materials and components are left to the Budget Allocation section.

5.2.2. The Solution

The electromechanical solution consists of the mobile base, corner angles that support a platform on which the extendible arm is installed, the vertical arm that is attached to the robot's body, and secondary attachments for sensors and the microcontroller board. The total weight of the machine is 2.3kg

The Mobile Base

The mobile base consists of:

- **(2)** 30cm Aluminum Angles (1" x 1" x 0.063")
- **(2)** 20cm Aluminum Angles (1" x 1" x 0.063")
- **(1)** 15cm x 15cm x 0.3cm Aluminum Sheet
- **(2)** 2" Rigid Caster Wheels
- **(2)** 25D mm Metal Gearmotor Brackets
- **(2)** 65:1 25D Metal Gearmotors (6V, 100RPM)
- **(2)** 90mm x 10mm Red wheels (90mm diameter, 10mm wheel thickness)
- **(2)** 4mm Aluminum Motor Hubs
- **(4)** 8-32 screws
- **(4)** M3 metric machine screws and nuts

Figure 5.2.2.1: The mobile base of the machine, with the absence of the aluminum sheet and shaft encoders.

The mobile base is a major design and construction component of the subsystem. Since the materials for items such as casters, screws, and motors are limited in variety, the overall materials selection primarily involves that of the base, framework, suspended platform, and the retractable arm. Aluminum was the material chosen for the base, the corner angles, the supports for the platform, and for the channel through which the arm retracts. A table similar to the following aids in the decision for material selection.

	density ρ (lbs/in³)	density ρ (kg/m³)	tensile strength (psi)	tensile strength (MPa)	Young's modulus E (psi)	Young's modulus E (GPa)	specific modulus (E/ρ)
aluminum	0.10	2700	10,000	70	12×10^6	80	1.2
brass	0.30	8500	60,000	400	15×10^6	100	0.5
copper	0.32	8900	22,000	150	17×10^6	120	0.5
fiberglass	0.065	1800	40,000	275	4×10^6	30	0.6
glass	0.094	2600	15,000	100	10×10^6	75	1.1
carbon composite	0.054	1500	400,000	2800	20×10^6	100-400*	3.7
Kevlar	0.049	1350	200,000	1400	12×10^6	80	2.5
Lexan	0.043	1200	8,000	55	0.3×10^6	2	0.1
nylon	0.041	1150	11,000	75	0.3×10^6	2	0.1
Plexiglas	0.042	1160	6,000	40	0.6×10^6	4	0.1
steel (mild)	0.28	7850	70,000	480	30×10^6	200	1.1
steel (piano wire)	0.28	7800	400,000	2800	30×10^6	200	1.1
wood (balsa)	0.008	215	2,000	14	1×10^6	7	1.3
wood (oak)	0.025	750	15,000	100	2×10^6	15	0.8
wood (pine)	0.015	410	6,000	40	1.5×10^6	12	1.0

Table 5.2.2.2: Properties of typical design construction materials. (Source: B. Davies, Practical Robotics, Toronto: WERD Technology Inc., 1997.)

However, many of the materials aside from wood and aluminum are either inaccessible, more costly, had a larger weight to volume density, or could not be utilized for ease of construction, such as glass or steel.

Wood is also used in different parts of the machine but, the design is such that the aluminum, which has a much higher density, is used where the machine is closest to the ground. The idea behind this design decision is to lower the centre of gravity so that the machine is better stabilized, and less prone to toppling over. Similarly, out of the two horizontal platforms that comprise the machine, the base is designed to be heavier than the one placed at the top, with the heaviest components being the two caster wheels, the two DC Gearmotors, the battery power supply, and the aluminum angles.

Motor Selection of the Base

A large variety of compact motor actuators exist on the market today, the majority of which can be used in small-scale applications. As such, a variety of factors must be considered for an informed and careful motor selection, but the most critical factors include the load capacity needed or the motor torque, the required rotational speed under the specified load, and the required voltage. [1]

Figure 5.2.2.2: 6V, 100RPM DC Gearmotor. Two were used in the construction of the base.

The motors selected for the mobile base are the Zhengke 65:1 25D Metal Gear Motors, with 100rpm rotational speed and 6V. Commonly used DC motors have a general inverse relationship between torque and speed. Torque, as it relates to the wheel motors, describes the motor’s ability to overcome the weight imposed by the body of the robot and the frictional forces opposing the turning of the wheel. The rotational speed refers to the speed at which the wheels can turn while carrying the load. This speed is reduced as the weight of the robot increases. As a result, these motors were selected over stepper motors or servo motors because DC gearmotors are the simplest to control from a circuitry perspective and provide enough speed required to traverse the row of barrels within the time constraint of 3 minutes.

The following table lists similar (not exact) DC gearmotors offered by Zhengke motors. A 100rpm motor was selected (not listed below) as it is typical rotational speed for 6V, the lowest voltage available for such motors.

Specification:

Motor Voltage	Motor Speed	Ratio	Gearbox Legnth	No load		At Max.Eff.			Max.Torque
VDC	rpm	i	mm	Current (A)	Speed (rpm)	Current(A)	Speed (rpm)	Torque(mN.m)	(mN.m)
6V	4700	654	27	0.20	7	0.45	5	850	1700
6V	4700	474	27	0.20	10	0.45	7	612	1225
6V	4700	215	24	0.20	22	0.45	15	278	555
6V	4700	156	24	0.20	30	0.45	21	202	403
6V	4700	113	24	0.20	42	0.45	29	146	292
6V	4700	98	22	0.20	48	0.45	34	127	253
6V	4700	60	22	0.20	78	0.45	55	78	155
6V	4700	51	22	0.20	92	0.45	65	66	132
6V	4700	32	19	0.20	147	0.45	103	41	83
6V	4700	23	19	0.20	204	0.45	143	30	59
6V	4700	15	19	0.20	313	0.45	219	19	39
6V	4700	11	19	0.20	427	0.45	299	14	28
12V	5600	654	27	0.12	9	0.23	6	777	1555
12V	5600	474	27	0.12	12	0.23	8	563	1127
12V	5600	215	24	0.12	26	0.23	18	256	511
12V	5600	156	24	0.12	36	0.23	25	185	371
12V	5600	113	24	0.12	50	0.23	35	134	269

Table 5.2.2.2: 25mm DC gearmotors offered by Zhengke. Taken from http://www.zhengkemotor.com/product/product20.html

Shaft Encoders for Mobile Base

To minimize deviations from a straight path traversal, a correction mechanism is implemented known as shaft encoding. The objective of shaft encoders is to count the distance travelled by each wheel, and by extension, the number of rotations of the motor shaft, and send this information to the processing and control microcontroller. This mechanism can assist in controlling the rotational speeds of the driving wheels, and thus allowing a greater control over the steering.

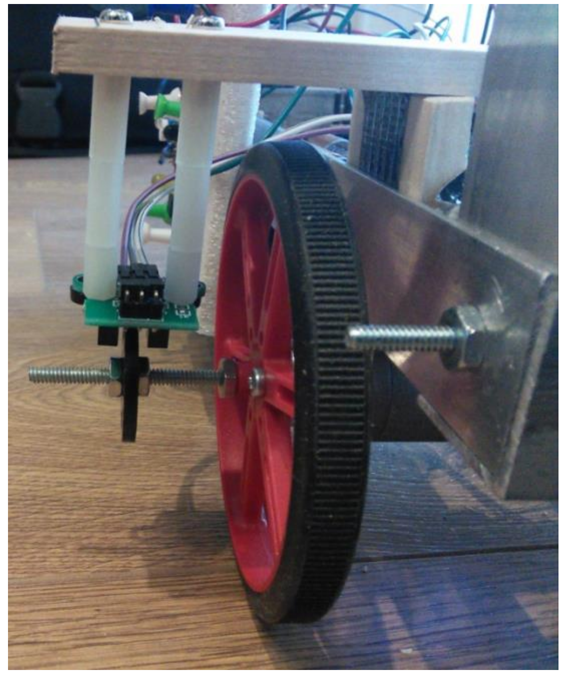


Figure 5.2.2.3: Optical shaft encoder secured onto threaded rod. The break-beam sensor is brought down using standoffs to count the number of slits passing through.

The shaft encoders installation consists of:

- **(4)** 15mm Nylon Standoffs
- **(4)** 10mm Nylon Standoffs
- **(4)** 6mm Nylon Standoffs
- **(2)** 4cm x 8cm Basswood (1/4" thick)
- **(2)** 5cm x 5cm Basswood (1/4" thick)
- **(1)** 6-32 1' Threaded Rod
- **(6)** 6-32 Nuts
- **(4)** M3 metric machine screws

The threaded rod was cut to two 6cm lengths to be inserted into the pre-made hole of each wheel. The hole was first tapped using a tap with a 6-32 thread, and a thread was carved through the red plastic. This was an extremely efficient technique shaft is easily removable from the wheel, if the encoder needs to be adjusted, but is secure enough that the encoder can rest with slipping across the thread.

Figure 5.2.2.4: Slits of the encoder visible from side view of wheel.

Arm Material and Mechanics

The material for the arm is basswood. A highly stiff wood with low density, it is able to extend 35cm while exhibiting almost unnoticeable deflections. The horizontal portion of the arm is 50cm x 3cm with a ¼” thickness. The vertical portion mounted onto the machine is 25cm x 3” with a ¼” thickness. The vertical portion attached to the horizontal arm using a ¼” corner brace is 28cm x 4cm with a ¼” thickness.

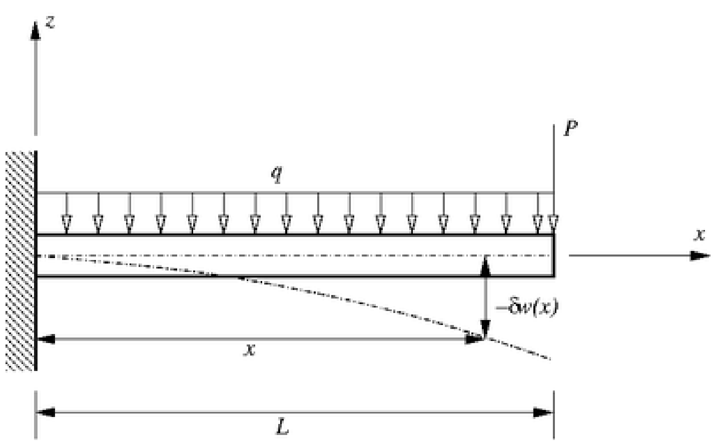
Mounted atop the horizontal arm are seven 2.5” long, green acetal rack gears made by Vex Robotics. Each weighs approximately 0.02lbs. These were chosen along with a green, acetal spur gear also from Vex to convert the rotational motion of the arm motor to the linear retraction and extension of the arm. Although there are various racks existing on the market, at a low cost and highly durable material, the final selection included these.

Figure 5.2.2.5: Vex Rack gears.

Rack Gear - (276-1957)

The platform that supports the arm consists of a 17cm x 28cm x ½” piece of plywood and two ¾” x ¾” x 20cm aluminum angles on the front and rear sides. A 6cm x 5cm block of pinewood of height 1.95cm is mounted adjacent to the arm track, with an aluminum bracket resting on top, securing the motor.

When extended, the arm acts as a cantilever beam with a linearly distributed load. However, since there exists a vertical portion of the arm at the end, the moment carried increases at a constant rate and terminates at the total moment exerted by both the horizontal and vertical arms. The following illustration demonstrates the mechanics principle behind the cantilever beam.



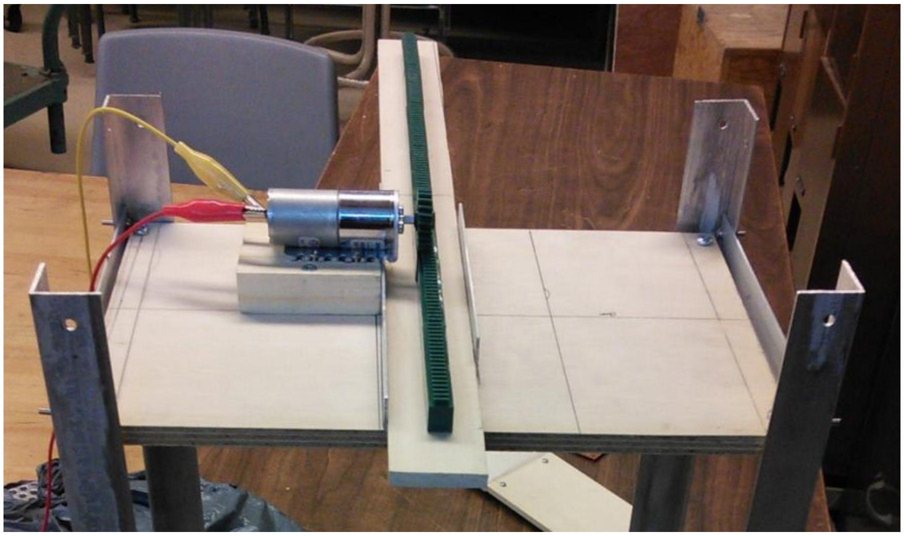


Figure 5.2.2.6: The top platform supporting the arm mechanism.

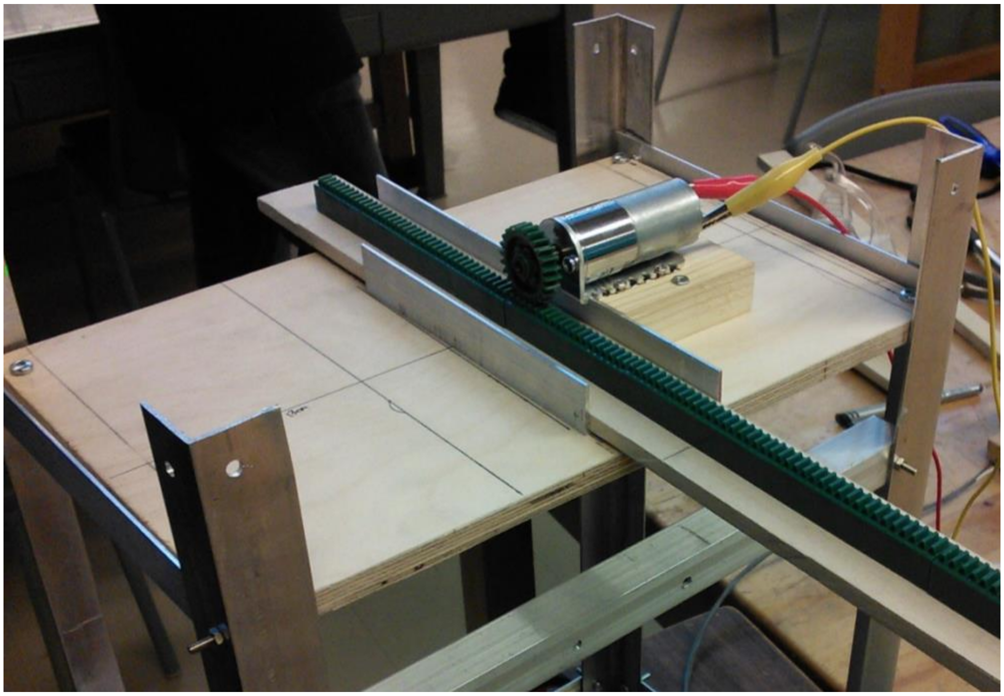


Figure 5.2.2.7: Different view of the arm arrangement.

5.2.3. Future Improvements to the Subsystem

Though all mechanisms of the subsystem are functional and it is able to accomplish all tasks as required by the electromechanical specifications of the problem, there many improvements that can be made to the design. The most important of these is the replacement of the caster wheels with another pair of the red 90mm x 10mm wheels. After many test runs, it was frustrating to see that the machine would deviate despite the shaft encoding and the numerous realignments of the wheels. By replacing the casters with the red wheels, all wheels would turn at the same rate, eliminating any differences in friction and surface effects.

5.3. Power, Instrumentation and Interfacing Subsystem

The Power, Instrumentation and Interfacing Subsystem was completed by Stefan Momic.

5.3.1. Assessment of the Problem

The instrumentation and interfacing subsystem deals with all the digital and analog interfacing electronics which connects the sensors and actuators to the microcontroller board. Furthermore, this subsystem deals with all the sensors and input/output signal calibration/protection with the major sensory task of this system involving shaft encoding and barrel and liquid level detecting. Lastly the subsystem is responsible for acquiring all the suitable power supplies for the actuators, circuits, sensors and microcontroller.

Due to the nature of the subsystem, the following design considerations needed to be addressed:

- Decide the kind and nature of connection between the sensors and actuators to the microcontroller board.
- Decide the kind and amount of sensors needed to perform each task (detecting the barrel, determining the water level of each barrel, detecting the column and measuring the distance travelled).
- Decide what kind of calibration/protection methods need to be in place to ensure smooth operation of the system.
- Determine the suitable power supplies needed for the actuators, circuits, sensors and microcontroller and determine the best way to implement them.

The assessment of the problem from a circuits' subsystem point of view include the need to implement the following tasks:

7. Travel in a straight line for a maximum of 4m and return to the 'Start Line'
8. Detect the location of a barrel and clearly signal its detection

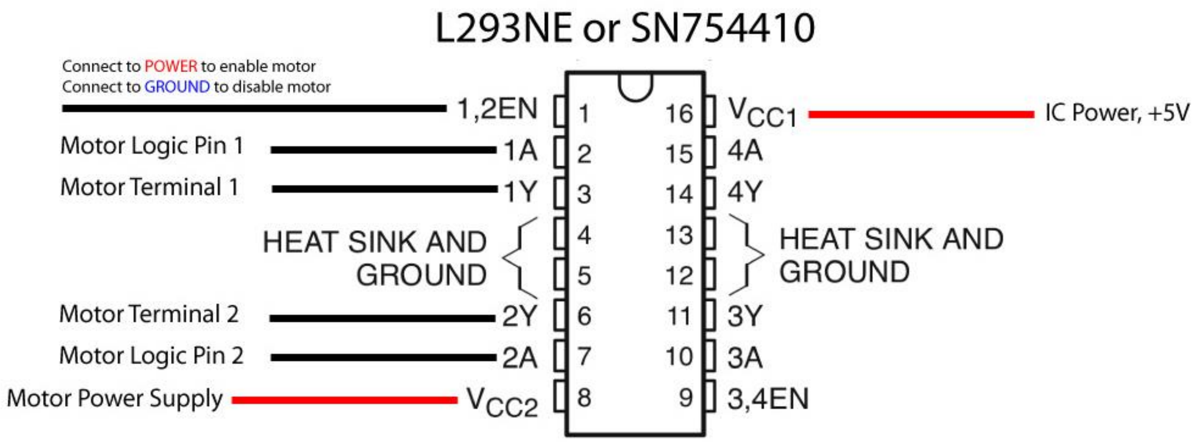
9. Determine the type of barrel (tall or short)
10. Determine the location of the barrel from the 'Start Line' to the barrel centerline.
11. Determine the level of heavy water in each barrel
12. Detect the location of the column and retracting the arm to avoid contact

5.3.2. The Solution

The solution to each of the tasks listed above is discussed in the sections below.

Task #1: Travel in a straight line for a maximum of 4m and return to the 'Start Line'

For the completion of this task, a sensor is needed to detect the machine starting to turn (or go off path) and send the necessary signals to the microcontroller to take the necessary actions to get the machine back on track. For the competition of this task, the Optical Encoder Set for DC Gear Motor from Creatron Inc. was selected. Other possible alternatives considered included other distance sensors and shift encoders, however the Optical Encoder Set was selected due to their ease of integration with the microcontroller. In hindsight, the difficulty of mounting these encoders provided a difficult task for the electromechanical member and in future projects an easier to mount encoder should also be considered. Distance sensors were eliminated due to the dependence of distance sensors needing a reference with which to gauge itself, shift encoders are clearly the better design decision for completing this task, since the barrels are not all in a straight line but are able to vary with a ±1 cm variance and are not always in a sensible range. Shift encoders work by monitoring the distance that each wheel travels and corrects itself if one wheel travels more than the other. Whenever one wheel gets ahead of the other, it causes the machine to turn. As soon as this difference in distance is determined, that wheel needs to be slowed down to allow the other one to catch up in order to ensure that travel occurs in a straight line. Wheels are slowed down by stopping them altogether using Pulse Width Modulation (PWM) signals sent out by the microcontroller. The motors to control the wheels are connected to a L293H H-Bridge whose pin schematic is shown in Figure 5.3.2.1 below.



EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

Figure 5.3.2.1: L293 H-Bridge Pin Port Diagram

For controlling the turning one of the motors off, either Pin 1 or Pin 9 are set to ground to disable the particular motor that has travelled the further distance, allowing for the motor that has travelled less to catch up.

Additionally shaft encoders are preferred for the performance of this task because they are further used for the completion of task #4 – determining the location of the barrel from the 'Start Line.'

It should be noted that wheel slip may be a limitation of shift encoders as they will record a distance travelled but not actually travel that distance. If only one wheel slips but not the other, then the robot will turn and ultimately direct itself in the wrong path. This limitation is ultimately addressed by ensuring that the machine travels at a slow enough speed and is heavy enough so that the wheels do not slip. We took this approach because we believe that the machine should go slowly due to safety concerns. The machine is at a maximum height of 55 cm which may be above a worker's field of vision. If the worker happens to be in the protected area and the machine crashes into him, ensuring that the machine travels at a low speed prevents serious injuries.

<u>Task #2: Detect the location of a barrel and clearly signal its detection</u>

An HC-SR04 Ultrasonic Sensor, placed at the front of the machine body at the bottom of our robot (roughly 9cm from the ground) was selected for the completion of this task. To signal the detection of the barrel it was decided that as soon as the barrel is detected a LED light is turned on. The LED light stays on for the duration of the barrels presence and turns off once the ultrasonic sensor stops getting the signal that was calibrated to mean barrel presence.

Wide arrays of proximity sensors are available for achieving such a task; however of these we narrowed our focus on infrared, ultrasonic and laser sensors. Other sensors such as the Eddy Current Sensor were eliminated from consideration due to their ability to only detect conductive objects [5]. Upon experimentation it was determined that infrared sensors are not feasible for the completion of the task due to the prospect of black objects absorbing too much of the incident infrared light. Experimentation results are tabulated in Appendix B, where it is evident that the voltage difference across the IR Receiver is too narrow a range when detecting the black barrel and ambient light to be confidently used as the barrel detection sensor. Laser sensors were also disregarded as a result, due to the belief that the black colour will also absorb much of the incident light since it is in the optical range of the electromagnetic spectrum [6].

As a result, ultrasonic sensors were the selected sensors for barrel detection due to their response independence on colors of object, despite the slightly higher cost. Limitations of the ultrasonic sensor include a slightly slower response time, false responses for loud noises and inability to detect objects of low densities. [7] Despite the limitations however, we believe that ultrasonic sensors are the perfect fit for the detection of barrels as the working environment will not be too loud, and the barrel is dense with good reflection of ultrasonic waves.

<u>Task #3: Determine the type of barrel (tall or short)</u>

An HC-SR04 Ultrasonic Sensor, placed 30cm from the ground is used to determine the type of barrel. The logic for this is simple: if the bottom sensor detects an object and the ultrasonic sensor that is mounted at 30 cm does not, then the detected object must be the shorter barrel. Alternately, if the top sensor also detects an object, then the barrel detected is the taller barrel. The sensor to be used is the same sensor as the one used to complete Task #2 due to the easy of controlling and programming the same sensor twice while only varying their location on the robot.

<u>Task #4: Determine the location of the barrel from the “Start Line” to the barrel centerline</u>

This task is mostly accomplished by the Optical Encoder Set for DC Gear Motor used for Task #1 and it’s the microcontroller’s job to save the distance recorded by the shaft encoders at the time that the barrel is detected.

As in task #1, wheel slip is a limitation of this approach; however it could be approached by the similar philosophy that a low speed and heavy robot would prevent wheel slip. A greater limitation is the fact that the distance from the 'Start Line' is in accordance to the centerline of the barrel. When testing the robot it was noticed that the ultrasonic sensor was able to accurately identify the barrel as soon as it was present. Knowing this, all that needs to be done to get the distance from the 'Start Line' to the centerline is adding the radius of the barrel to the measured distance obtained by the encoders.

Task #5: Determine the level of heavy water in each barrel

8 Infrared Sensor circuits in parallel are used to determine the level of heavy water in each barrel. Each circuit requires a LM/uA741 OP-AMP, 10k Potentiometer, 100 Ω resistor, 47 kΩ resistor, 330 Ω resistor, IR Emitter, IR Receiver and an LED light (yellow for the short barrels and green for the large barrels). The logic for determining whether the barrel is full, half-full or empty is quite simple and is displayed in Figure 5.3.2.2 below. Determining the level of water is achieved by the sensors S_1, S_2, T_1, and T_2 which are meant to be able to detect white. It is noted that sensors S_1 and S_2 are placed at 1/3 and 2/3's the height of the short barrel at 8.33cm and 16.66cm from the ground while sensors T_1 and T_2 are placed at 1/3 and 2/3's the height of the tall barrel at 11.83cm and 23.66cm from the ground. The logic for determining the water level is discussed in the Table 5.3.2.1 below.

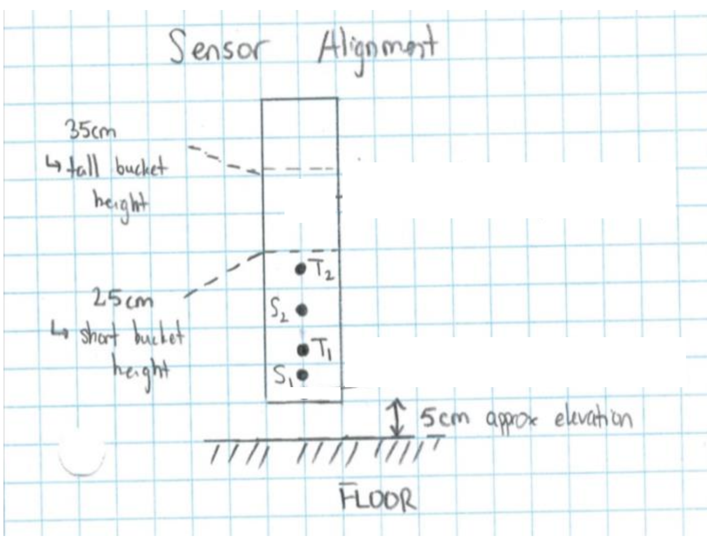


Figure 5.3.2.2: Alignment of the sensors on arm

Sensor S/T 1	Sensor S/T 2	Result
0	0	Empty
1	0	Half-Full
1	1	Full
0	1	Not-Possible

Note: 0 represents not detected and 1 represents detected.

Table 5.3.2.1: Water Level Sensor Logic

It should also be noted, that the second ultrasonic sensor (the one located at 30cm from the ground) acts as a multiplexer to select which sensors are used to determine water level – if the ultrasonic sensor detects an object, the T sensors are used; else, the S sensors are used. Note that the same sensor logic is also used on the body of the robot in case the water level indicator bar is in that orientation.

In summary, 8 sensors are required, 4 of which will be on the inside of the arm, and the other 4 on the robot's body. This is due to the fact that the level indicator will either face the robot or exactly 180 degrees away from the machine with a reasonable range of uncertainty. Of these 4 sensors are on each side, 2 are for the tall barrel and 2 are for the short barrel. The set of sensors to be used is determined by whether or not the ultrasonic sensor placed at 30cm detects an object or not. The circuit diagram of the IR Sensor circuit, along with a picture of the real circuit are displayed in Figures 5.3.2.3 and 5.3.2.4 respectively.

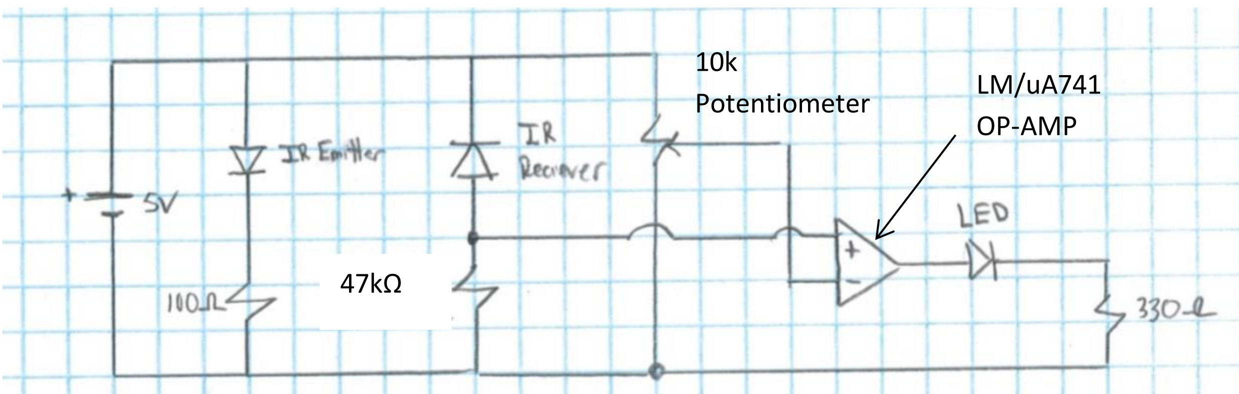


Figure 5.3.2.3: IR Sensor Circuit Schematic Diagram. Note that the 5V source is from the VCC Port of the PIC DevBugger Development Board

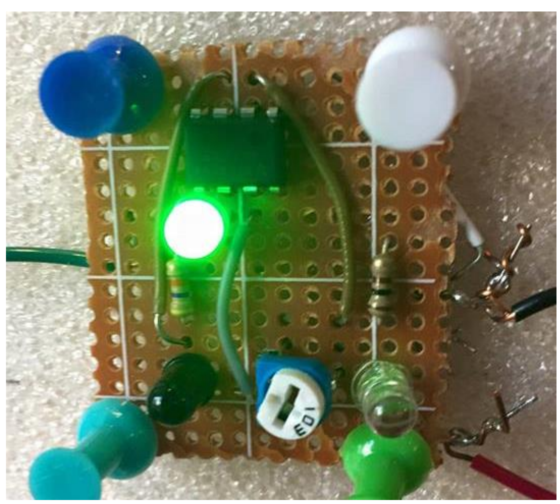


Figure 5.3.2.4: IR Sensor Circuit

Prior to selecting IR Sensors for the completion of this task, photoresistors were also considered. Experiment results are summarized in Appendix B. Due to the greater voltage difference in comparing black and white using an IR Sensor, calibrating the threshold value will be easier and have greater certainty. As a result, IR Sensors were selected are the method of choice for determining the water level in each barrel.

Task #6: Detect the location of a column and retracting the arm to avoid contact

For the detection of the column, an HC-SR04 Ultrasonic Sensor is placed in the front of the machine body, in line with the other 2 ultrasonic sensors at a height greater than the tall barrel (approximately 40cm from the ground). The same sensor for detecting the barrel will be used for detection of the column since they are the same colour and it's easier to implement the same sensor in a different location having done so already elsewhere. As a result, when this sensor detects something it must be the column since it is placed too high to detect either of the barrels. Upon detection, the microcontroller makes the machine stop by disabling both of the drive motors and enables the H-Bridge pin that is responsible for controlling the 12V Zheng DC Gearhead Motor, which retracts the arm. After retracting the arm, the wheel motor controllers are enabled driving the machine forward until the column is cleared at which point the robot stops again and extracts the arm back to the original position.

5.3.3. Supporting Calculations

A more detailed summary of supporting calculations is provided in the Engineering Design Handbook. However, a summary of the important calculations are provided below.

Infrared Sensor Placement

Tall Barrel: $35.5cm$

$$35.5 \div 3 = 11.8\overline{3}$$

- ➔ Sensors need to be placed at $11.83cm$ and $2 \times 11.83 = 23.67cm$ from the ground.
- ➔ Since the arm and side robot panels was designed to be $3.2cm$ from the bottom, these sensors needed to be placed $11.83 - 3.2 = 8.63cm$ and $23.67 - 3.2 = 20.47cm$ from the bottom of the basswood.

Short Barrel: $25cm$

$$25 \ \div 3 = 8.\overline{33}$$

- ➔ Sensors need to be placed at $8.33cm$ and $2 \times 8.33 = 16.67cm$ from the ground.

- ➔ Since the arm and side robot panels was designed to be $3.2cm$ from the bottom, these sensors needed to be placed $8.33 - 3.2 = 5.13cm$ and $16.67 - 3.2 = 13.47cm$ from the bottom of the basswood.

Determining the Resistance of the Motor

This calculation was desired in order to obtain the optimum resistor to be placed before the arm motor after realization that it was drawing too much current (so much so that the microcontroller could not turn on anymore).

Motor: $12V$

Current

- ➔ No Load: $0.12A$; Load: $0.23A$

Using Ohm's Law, as stated in the equation below,

$$R = \frac{V}{I}$$

it was calculated that the resistance of the motor was 100Ω (no load) and 52.17Ω (load).

It was noted that despite changing the motor to 6V in order to meet the budget constraints, putting a 51Ω in series with the H-Bridge did indeed work and it enabled both our H-Bridge motor and microcontroller to work at the same time.

Power Requirements Calculations

The key formula of use for this section is $P = VI$ where P is power in Watts [W], V is voltage in volts [V] and I is current in amperes [A].

Component	Voltage [V]	Current [A]	Power [W]
IR Emitter and Receiver (LTE-4208)	5	0.05	0.25 Since we need 8 of these, the total power consumption of the IR Sensors is 2W
HC-SR04 Ultrasonic Sensor	5	0.015	0.075 Since we used 3 of these, the total power consumption by the

			Ultrasonic Sensors is 0.225W.
PIC Microcontroller	12	1.5	18
Shift Encoder	5	<0.5	2.5 Since we need 2 of these, the total power consumption is 5W

Table 5.3.3.1: Power Requirement Calculation Summary

By looking at the power specifications, it was decided that two different batteries will be used. One 6V battery to power all of the 5V circuits, and one 12V battery to focus on powering the microcontroller. It should be noted that this was later changed once more information regarding the motors was known (at the time of calculation it was thought that the GM8 143:1 GEAR MOTORs were going to be used). When the motors were changed to be 6V Zheng DC Motors, it was decided to have one 6V battery used to power all the motors, while a 12V battery powered the microcontroller board which in turn provided power for the IR Sensors, Encoders and Ultrasonic Sensors.

5.3.4. Potential Subsystem Improvements

A lot of potential improvements could be said regarding subsystem improvement particularly since the system failed to qualify a run in the competition. A major improvement for the system is regarding wiring that is more organized making it easier to debug. Having such an organizing wiring would ensure that no wires open or short circuited and eliminate the time spent debugging only to find that it was one of the connections that became undone. Doing this would also create a more moduized framework that would enable easy replacement of parts if the need be, which eliminates the need to redo an entire circuit just because on item went wrong.

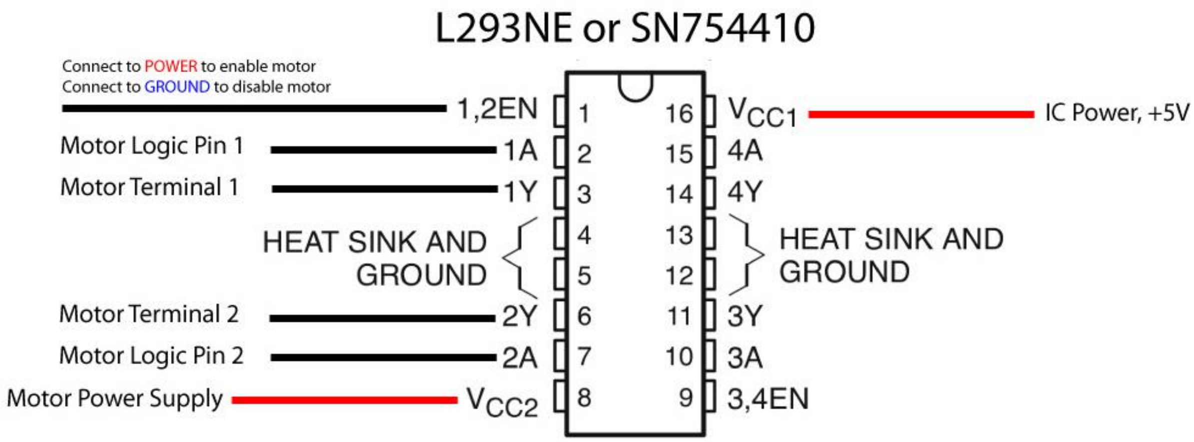
Apart from this, unfortunately not much else can be said regarding improvements to the IR Sensor circuits. The encoders and ultrasonic sensors seemed to work fine and considering that a lot of groups used the same ones, I would not change them if the opportunity to redo the project arose. The IR Sensors were never actually used except for the Week 10 Evaluation due to the priority of getting the robot to drive straight and detect the barrels/columns. However, due to the ease in which we got them to interact with the microcontroller for Week 10, I feel as though they were well done. The use of the potentiometer in the circuit was extremely useful especially since it was easy to calibrate just by turning the potentiometer until the desired threshold voltage was obtained.

6. Integration

Subsystem integration began on February 22, 2016 after the Week 8 Evaluation. Unfortunately, the Week 8 deliverable by the Electromechanical member (which was completion of the entire machine platform and mechanisms) was not completed so he had to continue working on his subsystem to catch up. The Electromechanical subsystem was completed by March 10, 2016.

In the meantime, the Microcontroller and Electromechanical member worked diligently on the subsystem integration. Figuring that the drive mechanism was the most important, they started working on getting the H-Bridge to work and be able to be controlled by the microcontroller board. It was decided to use an H-Bridge IC Chip to control the drive motors since it was small and compact, easily fitting in the inside of the machine. Knowing that one open H-Bridge was needed, it was decided to implement the open H-Bridge on the top of the machine and use it for the Arm motor.

An L293 quad H-Bridge was selected as the H-Bridge IC Chip to control the drive motors and was connected according to the schematic shown in Figure 6.1 below. The motor power supply pin (PIN 8) was connected to the 6V power supply. PINS 4, 5, 12 and 13 were grounded to the common ground on the main robot protoboard. The power for the H-Bridge (PIN 16) was provided from the VCC pin on the Microcontroller board. The motor terminal PINS 3 and 6 were used for the right hand motor (right being defined from the robot when the microcontroller board is oriented towards you), while PINS 11 and 14 were used for the left hand motor. The remainder of the pins were connected to the microcontroller board and were used as Motor Logic Pins. PINS 1 and 9 were connected to ports C0 and C1 on the microcontroller board. These pins were responsible for enabling the right and left motors respectively. PINS 2 and 7 were the motor logic pins for the right hand motor and were used to control the direction of the motors rotation, the logic of which is displayed in Figure 6.1 below. PINS 2 and 7 were connected to ports C2 and C5 respectively. Similarly PINS 10 and 15 were the motor logic pins for the left hand motor and followed the same logic as the right. These pins were connected to ports C6 and C7.



EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

Figure 6.1: H-Bridge IC Schematic Diagram

As stated previously, the main purpose of having an H-Bridge was to control the direction and speed of the motors. This was needed for two main purposes as described in the points below:

- Ensure the machine drives straight by tracking how far each wheel travels and stopping one of the motors to allow the motors to travel the same amount. If each wheel travels the same distance, this in turn means that the machine drives straight (neglecting any wheel slip, which did not seem to occur).
- Enable the machine to travel backward upon completion of 7 barrels inspected or 4m travelled.

The H-Bridge was interfaced and tested on the GM8 143:1 GEAR MOTORS provided in the Project Kit to ensure that the logic worked and the microcontroller was indeed able to control the speed and direction of each motor. Unfortunately, testing whether the H-Bridge worked on our motors, or seeing if the machine was able to go straight was not possible prior to week 10 due to the electromechanical member needing the base so he could level it and install the arm/arm motor. Ultimately, this two week testing deficiency made going straight a real problem and did not leave the team enough time to test and debug the machine prior to the Project Demonstration. It is believed that with two more weeks would enable to machine to work perfectly, which is coincidentally how much time was lost due to Subsystem delays.

Additionally, during the first two weeks of integration in the absence of the base, the Circuits and Microcontroller member were able to interface the IR Sensors to communicate with the microcontroller board. The circuit was connected as depicted in Figure 6.2 below.

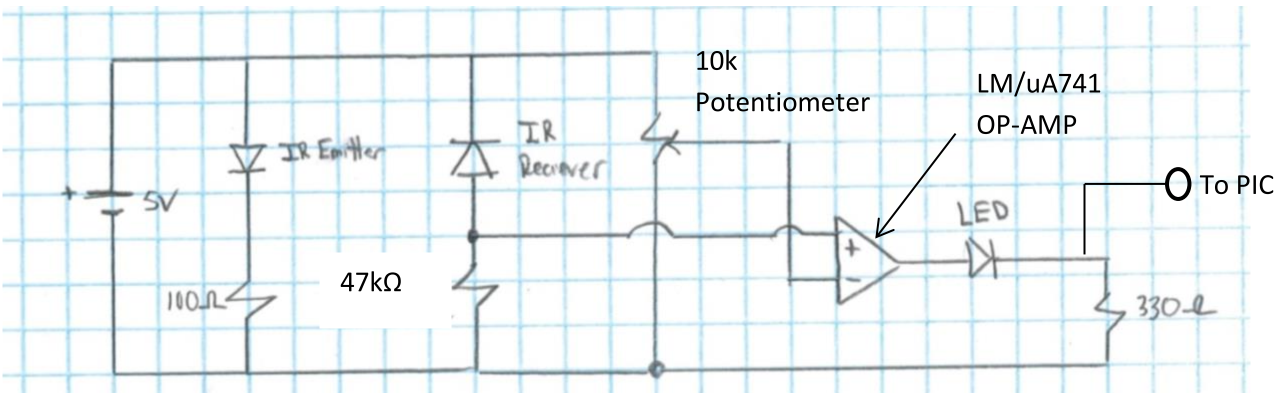


Figure 6.2: Interfaced IR Sensors schematic diagram.

The signal was connected to the PIC after the OP-AMP and the voltage readings were read by the microcontroller board. Unfortunately, prior to Week 10, the arm was yet to be created, so only the sensors to be attached to the robot were able to be created. For these 4 sensors it was tested that the voltage difference between the PIC node and ground was greater than $2.5V$ when white was detected and close to $0V$ when black was detected. As a result, code was created to say that a white was detected when the signal voltage read by the microcontroller was greater than $1.0V$. This was demonstrated to the TA during the Week 10 evaluation by having the LCD display the number of the sensor that detected white. Unfortunately, the logic for the level of water detected was not implemented in the main code due to the prioritization of getting the machine to drive straight and detect the barrels/columns, which was necessary for qualification. Once the arm was made, 4 more of these identical circuits were made for the arm of the robot. However, for this side, the voltage difference of the green LED IR Sensor Circuits (used for the tall barrels) gave a voltage reading of $0.8V$ when white was detected, while the yellow LED IR Sensor Circuits (used for the short barrels) gave a reading of $1.8V$. As a result, for this side of the arm, a threshold voltage of $0.5V$ was selected with which the logic was as follows: if the voltage reading is greater than $0.5V$, white tape was being detected, else it was black. Ultimately, 8 identical IR Sensor circuits were used, with 4 of them having green LED lights for the tall barrel, and 4 having yellow LED lights for the short barrel. Two of each colour were placed on the arm and body of the robot respectively. As a result, there were 8 microcontroller ports used by the IR Sensor circuits. The port each sensor was connected to is displayed in Table 6.1 below.

Sensor	Port
Body Sensor at 2/3 Height for Tall Barrel	A7
Body Sensor at 1/3 Height for Tall Barrel	A5

Body Sensor at 2/3 Height for Short Barrel	A4
Body Sensor at 1/3 Height for Short Barrel	A6
Arm Sensor at 2/3 Height for Tall Barrel	C3
Arm Sensor at 1/3 Height for Tall Barrel	B4
Arm Sensor at 2/3 Height for Short Barrel	C4
Arm Sensor at 1/3 Height for Short Barrel	A0

Table 6.1: PIC Port breakdown for IR Sensors

Additionally, it should be noted that the IR Sensors were powered by the VCC port of the PIC board. The 8 IR Sensor power lines were connected to the common VCC node on the main protoboard (along with the VCC from the H-Bridge IC power supply). Additionally, all 8 IR Sensor Circuits were grounded to the common ground with the H-Bridge IC.

After week 10 the base was complete, which lead to tests being able to be run which would determine whether or not the machine was able to go straight. Realizing that the machine does not travel straight and noticing that the wheels seemed to oscillate left and right great attempts were made to try to mount the wheels better. Although the wheels were fixed slightly, resulting in the robot to deviate less, other methods needed to be implemented. At this time the optical encoders were connected and mounted to the base, which would enable the microcontroller to read how much each wheel rotated. No additional circuitry was necessary to interface the encoders. The VCC pin of the encoder was connected to the same VCC as everything else. However, since the main solder board was becoming rather congested, and the PIC board had 2 other ground pins, it was determined to ground both encoders together, along with the ultrasonic sensors and connect the two encoders and three ultrasonic sensors on their own separate common ground. As a result, the biggest integration issue regarding the encoders was making sense of their outputs and attaching it to the machine. The attachment of the encoders to the machine proved to be a problem due to their very small and precise mounting spaces. In the end, the encoders were glued down to the base creating an overhang as displayed in Figure 6.3 below.

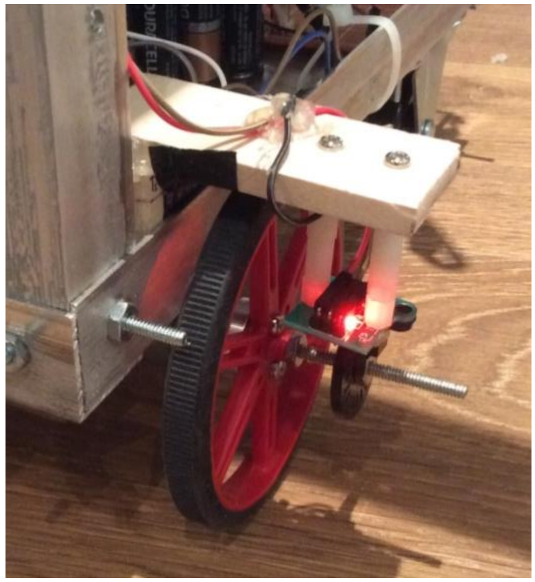


Figure 6.3: Encoder Mounts

The encoder outputs were connected to port B0 for the right hand wheel and B2 for the left hand wheel. Extensive testing was done from Week 10 until the Project Demonstration attempting to get the robot to drive straight. Blind Pulse Width Modulation (PWM) was also considered in which the motors were modulated without receiving feedback from the encoders in the attempt to achieve straight driving. By the time Project Demo came about, the robot was able to go fairly straight over a reasonable distance; however the correction mechanism was still subpar.

Along with the focus on the encoders to get the machine to drive straight, there was great focus on the ultrasonic sensors. Driving straight and being able to detect the barrels and column were singled out as being the most important tasks. The reason for this is because in order for qualification 3 barrels needed to be detected and none of them could be moved. For this, the encoders were essentially in driving straight and the ultrasonic sensors were needed to detect the barrels. Additionally the ultrasonic sensors were needed to detect to column in order to get the arm to retract. As with the optical encoders, no additional circuitry was needed for the Ultrasonic Sensors and they simply needed to be connected to the microcontroller board as displayed in Figure 6.4.

Figure 6.4: HC-SR04 Ultrasonic Sensor

The ground pin of all 3 ultrasonic sensors used, as labelled Gnd in the figure above was connected to the same ground node as the optical encoders which was connected to a GND port on the PIC board. The power for all three ultrasonic sensors, as labelled VCC was connected to the VCC of the microcontroller board through the main VCC node on the main solder board. Additionally, the triggers of two of the ultrasonic sensors (the one used for detecting the barrel located 5cm from the ground and the other used for determining whether it was a tall or short barrel located 30cm from the ground), labelled Trig in Figure 6.4, were connected in parallel to port E0. The third trigger, the one from the ultrasonic sensor located 40cm from the ground and used to detect the column was connected separately to port E1. Each echo, labelled Echo in Figure 6.4, was connected separately to the ports displayed in the Table 6.2 below.

Sensor	Port
Echo of the Ultrasonic Sensor 5cm from the ground	A1
Echo of the Ultrasonic Sensor 30cm from the ground	A2
Echo of the Ultrasonic Sensor 40cm from the ground	A3

Table 6.2: PIC Port Breakdown for the Ultrasonic Sensor Echo's

The ultrasonic sensors were mounted to the robot by gluing them on a piece of basswood, which was in turn glued to a bracket that was screwed onto the frame of the machine as displayed in Figure 6.5.

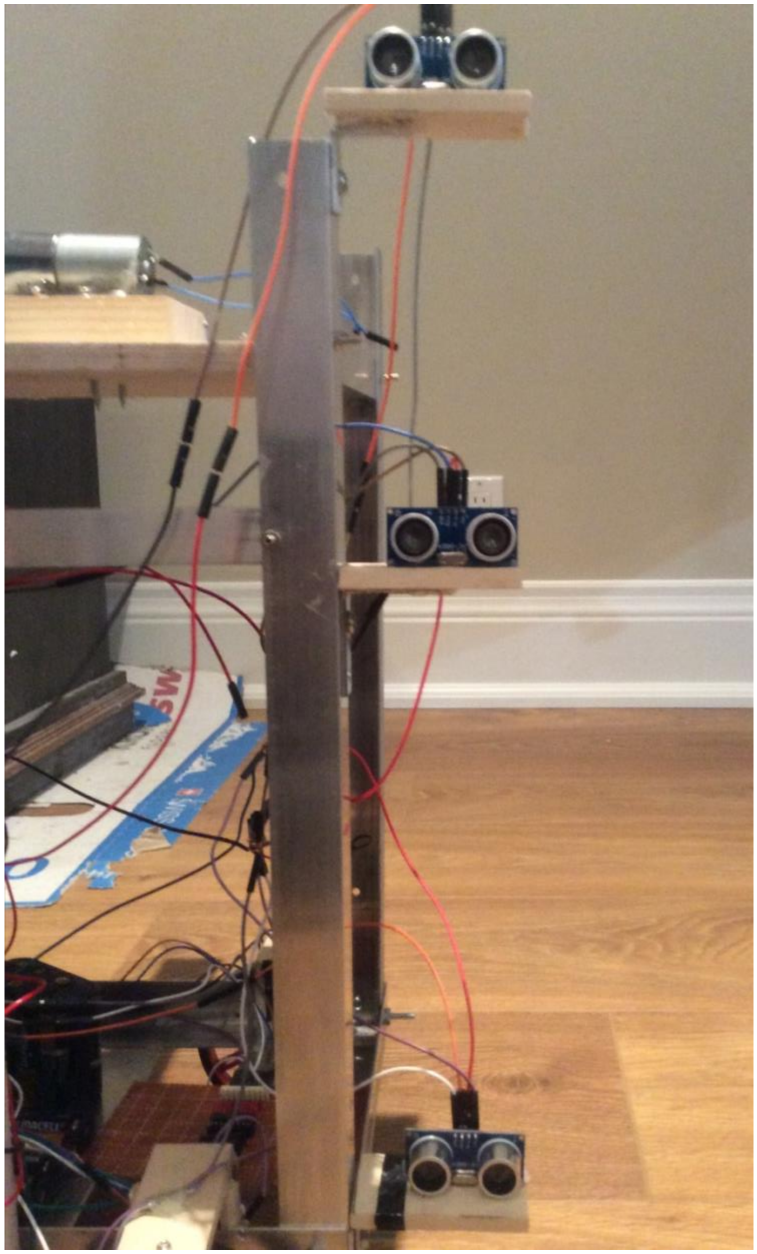


Figure 6.5: Ultrasonic Sensor Placement. Bottom sensor is at 5cm from the ground and is used to detect the presence of a barrel. The middle sensor is 30cm from the ground and its detection is used to signify the presence of a tall barrel. The top sensor is used to detect the column as it is placed higher than both barrels.

It should be noted that the location of the ultrasonic sensors changed from the original plan, in which it was proposed to have the ultrasonic sensors mounted together with the IR Sensors and the column detecting ultrasonic sensor mounted on the arm. The barrel detecting ultrasonic sensors were changed upon realizing that they needed to be at least 3cm from the object that needed to be detected. Having them on the arm was too close for them to make useful readings.

As a result they were moved to the front of the barrel where they could be at a reasonable distance from the objects that needed to be detected. With this, it was also noticed that as soon as the ultrasonic sensors went from detecting the barrel to not detecting the barrel the IR Sensors were in line with the water level indicator tape. This ultimately enabled the logic for getting the robot to stop in the right place relative to the barrel to be a lot easier and enabled for a calibration factor of safety. To signal the detection of the barrel, two LEDs were connected in series to port B3 that would be turned on whenever the bottom ultrasonic sensor detected an object. The Ultrasonic sensor that was planned to be on the arm was changed due to the difficult to mount it on the arm due to the rack. It was a lot easier to mount it on the robot body and seeing that it worked upon testing it gave us no reason to attempt a harder mount. The robot detecting the column and retracting the arm was demonstrated during Week 12's lab. It should be noted that prior to this time, an L293 H-Bridge was used for the Arm motor too. This was done because we already knew how to use the H-Bridge and it made debugging easier knowing that the H-Bridge for sure worked. Once the column was able to be detected, and the arm was able to retract it was apparent that the code worked and the Circuits member made the open H-Bridge. Any problems with the code from there would imply a problem with the H-Bridge and not the ultrasonic sensors or the code itself.

The H-Bridge was created according to the schematic presented in Figure 6.6, with the finalized H-Bridge presented in Figure 6.7.

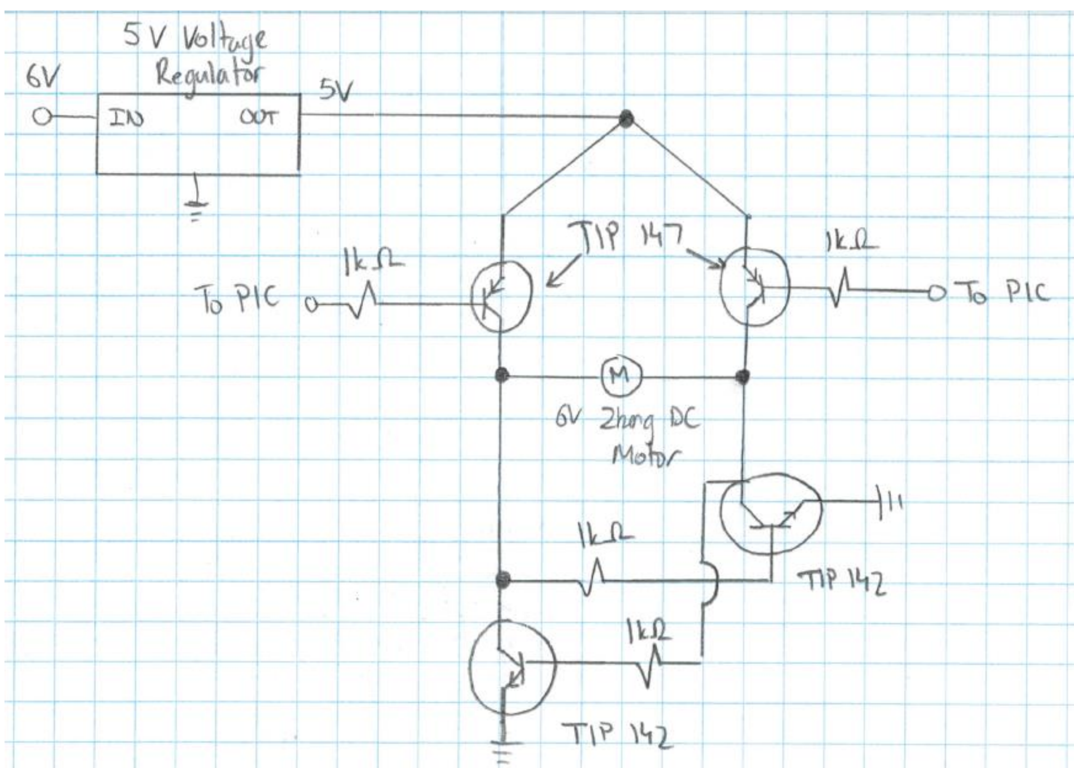


Figure 6.6: H-Bridge Schematic Diagram

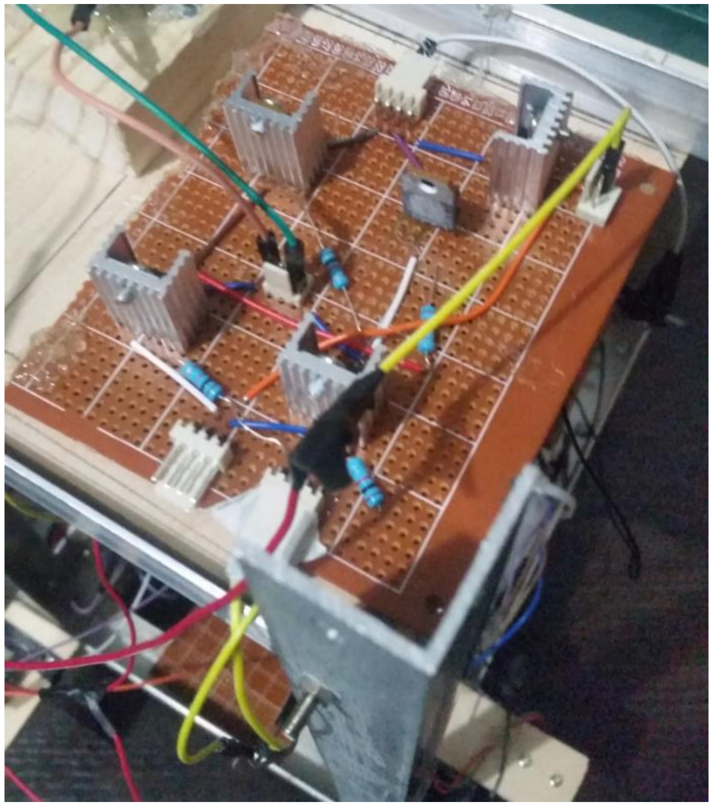


Figure 6.7: H-Bridge

The open H-Bridge was connected to a 5V voltage divider which was connected to the 6V power supply. Two direction selection wires came from the microcontroller, which were connected in ports D0 and E2. It should be noted that this was an active low H-Bridge which meant that it was turned off by having both motor controller pins set to high, and the direction was chosen by having one pin high and the other low. Setting both motor controller pins to low must never occur. Problems with the open H-Bridge arose due to the fact that initially we were using a 12V motor being powered by the 12V source. When the open H-Bridge circuit was created, the motor drew too much current and the microcontroller was not able to turn on. To reduce the current we put a 51Ω resistor in series with the H-Bridge motor. A 51Ω resistor was selected because the motor had a 52Ω load resistance. Connecting the equivalent resistance was desired because this maximized the voltage provided to the motor. This was done because it was observed that any other resistance was enough to power the microcontroller board, but was too big to power the motor. Despite this working, the design was further changed to use the 6V source on a 6V motor as the 6V motor was cheaper and we realized that the motor controller pins should provide the same voltage that was needed to run the motor. Since the microcontroller could only supply 5V it was figured that a 6V motor is preferred since powering a 12V motor with 5V was not enough. Having this design change was also beneficial in getting the machine under budget. The final design of which is presented in Figure 6.6.

6.1. Potential System Improvements

Due to the fact that the system did not end up qualifying, a lot should be able to be said about potential system improvements. The major reason for system failure however was not necessarily due to the system design or material selection as the materials used and system design was pretty standard in comparison to the other teams. The greatest downfall for our system was the lack of time available for System Testing and System Debugging due to certain subsystem deliverables not being met on time. Ideally, the deliverables would have been met on time, and if this was not feasible at least they would have been met closer than 2 weeks later.

Mounting the encoders is something that took too long and could potentially be worked on. However it is believed that the encoders took 2 weeks to be mounted due to a lack of effort or as an excuse to find something easier since it was easily mounted once all the members got involved. That being said however, an alternative design could be used for the optical encoders which would enable them to be mounted easier and save on some time.

Additionally better organization of the system in terms of wiring would be preferable for future projects as it would enable for easier debugging and prevent issues in which one of the wires would become open circuited. This would eliminate the scavenger hunt of where's the open wire, expediting the process of debugging by ensuring that any problems are actual problems and not simply problems with the wiring.

7. Schedule

This section includes a reflection on the Division of the Problem; finalized Gantt Charts which include the progress of the project and reflections on each member's week by week goals, deliverables and justifications for not meeting deliverable deadlines.

7.1. Division of the Problem

Much of the preliminary planning, including the conceptual design, system analysis and project planning has been performed through a close interaction of all three members of the team. However, the implementation of tasks has been broken up into three distinct subsystems – Microcontroller, Circuits and Electromechanical – with each member becoming a specialist of their assignment. The Microcontroller member is Amr Mahmoud; the Circuits member is Stefan Momic and the Electromechanical member is Hamza Nisar.

The Microcontroller member was responsible for the processing, control and communication of the system. This member programed all the software for the system which included all the combinational and sequential logic required to run the algorithms necessary for the completion of the task. Furthermore this member was in charge of designing the keypad and display interface with the microcontroller. Although initially, it was assessed that the subdivision of labour for this subsystem was significantly less; thereby adding the additional Design Features to be accomplished by the Microcontroller member, most of these were not accomplished due to the fact that a lot of the planned deadlines were not met. Of the originally planned Extra Design Features which included Real-time Date/Time Display, Permanent Logs, PC Interface and Remote Operation, only the Real-time Date/Time Display has been attempted to be completed. Furthermore, it was initially planned for the Microcontroller member to assemble a custom microcontroller board thereby allowing more of the budget to be used by the electromechanical and circuits members this was not completed due to the need of the Microcontroller member to assist the other members in completing their subsystem deliverables. The microcontroller member was able to deliver programmable code for basic design features by the time school resumed on February 22, 2016, however system integration did not begin until February 29, 2016 due to delays with other subsystems particularly that of the Electromechanical member. While the Microcontroller member waited to begin system integration he provided limited assistance in the fabrication and testing of solder board during laboratory hours. Once system integration finally began, the Microcontroller member became an integral member in the integration and debugging of the system.

The Circuits member was accountable for constructing all the digital and analog interfacing electronics which included connecting the sensors and actuators to the microcontroller board. Additionally, he was responsible for all sensors and input/output signal calibration and protection. Furthermore the Circuits member was required to work meticulously with the Microcontroller member for the parts of the circuits that are positional in nature such as the location of the sensors and stop switch. Additionally the Circuits member was able to design and build all the open circuits, including an open H-Bridge to avoid the use of the driver board provided in the Project Kit thereby saving costs and putting them towards the Electromechanical member and higher quality sensors. The major sensory tasks of this subsystem involved shaft encoding to calculate the distance travelled and ensure straight travel; and barrel and liquid level detecting. As a result, the Circuits member was responsible for the configuration and calibration of all the sensors and was expected to perform tests on them to ensure that they are able to consistently perform under their conditions of use. It is expected that the entire subsystem was entirely complete and functional by February 17, 2016 to ensure that system integration can begin as soon as Labs reopen on February 22, 2016. The Circuits member was mostly able to deliver for the deadline with the exception of providing an open H-Bridge due to the fact that he was informed of the finalized motor selection by the Electromechanical member a few days before the deadline. He was however able to compensate for this by providing an H-Bridge IC Chip which he knew would be sufficient for running on a wide range of possible motors chosen for the task. Once the lab reopened the Circuits member worked with the Microcontroller member to interface the IR Sensor circuits, ultrasonic sensors and H-Bridge IC Chip with the microcontroller. Much of this was performed while the Electromechanical member caught up with his Week 8 deliverable and the Week 10 Progress Evaluation displayed the work of the Microcontroller and Circuits member, without input from the Electromechanical member. After Week 10, the Electromechanical member finally caught up and system integration finally included all three members. Much of the preliminary testing of the machine was achieved using the H-Bridge IC Chips due to the importance of getting the code to work. Once the machine was able to detect the column and retract the arm, the Circuits member began construction of an open H-Bridge. This H-Bridge was then tested and implemented in the machine. Ideally this H-Bridge would have been completed during week 10 and 11 optional (Friday) labs. Unfortunately, the Microcontroller member was unable to attend due to the need to babysit kids in Orangeville. As a result, an open H-Bridge was not completed until week 13, but it was delivered prior to the Project Demonstration. During system integration, the Circuits member spent most of his time attaching the circuits to the frame and soldering wires of the right length to fit the design dimensions. He also helped debug code and ran tests to determine threshold values of the various sensors to enable them to be configured to the microcontroller's logic. Closer to the deadline of the Project Demonstration he put together the brochure and did majority of the final report due to the fact that most of the circuits were finalized and soldered and it was an issue with the code more than anything. During this time, the Circuits member helped the Microcontroller run any tests that were needed while working on the Final Report during the free time.

The Electromechanical member was responsible for constructing the platform, structure and frames and incorporating all the actuators and mechanisms required by the system. Furthermore, the Electromechanical member was responsible for the design and fabrication of the mobile platform, structure and frames, barrel detecting, and liquid measuring mechanisms. With cooperation between the other members, particularly the Circuits member, the Electromechanical member was responsible for assigning the locations of the sensors and boards. The Electromechanical member was required to have the machine mobile base complete and display full functionality of the wheel actuators by February 1, 2016 with the completed subsystem ready by February 17, 2016. Although he was able to provide the bottom frame of the base by February 1, 2016 the same base was presented for the Week 8 evaluation (on February 22nd) with the addition of 4 brackets aluminum brackets installed, one in each corner. The arm was not completed until after week 10 which is why only 4 IR Sensors were presented for Week 10's evaluation, which was done without the Electromechanical member's contribution. With the aid of the other members, the Electromechanical member was also expected to play an integral role in the integration of the subsystem starting February 22nd, however this was not the case due to his focus on finishing his own week 8 deliverables up until week 10. After completion of his week 8 deliverables in week 10, he assisted the other members in assembling the machine and testing.

7.2. GANTT Charts

The Gantt Chart as presented in the Project Proposal is repeated, along with the progress bars and rolled up milestones.

TASK	WEEKS	COST	ENG.
A	2	$0	MAH/MOM/NIS
B	3	$10	MAH/MOM/NIS
C	1.5	$40	MOM
D	3	$50	NIS
E	2	$0	MAH
F	1	$15	MOM
G	1.5	$0	NIS
H	1.5	$35	NIS
I	3	$35	MOM
J	3	$25	MOM
K	3	$0	MAH
L	3	$0	MAH
M	3	$40	NIS
N	3	$15	MOM
O	3	$50	NIS
P	2.5	$25	MAH
Q	2	$10	MOM
R	1.5	$10	MAH/MOM
S	3	$10	MAH
T	3	$15	MAH/MOM
U	2.5	$20	MAH/MOM/NIS
V	2	$15	MAH/MOM/NIS

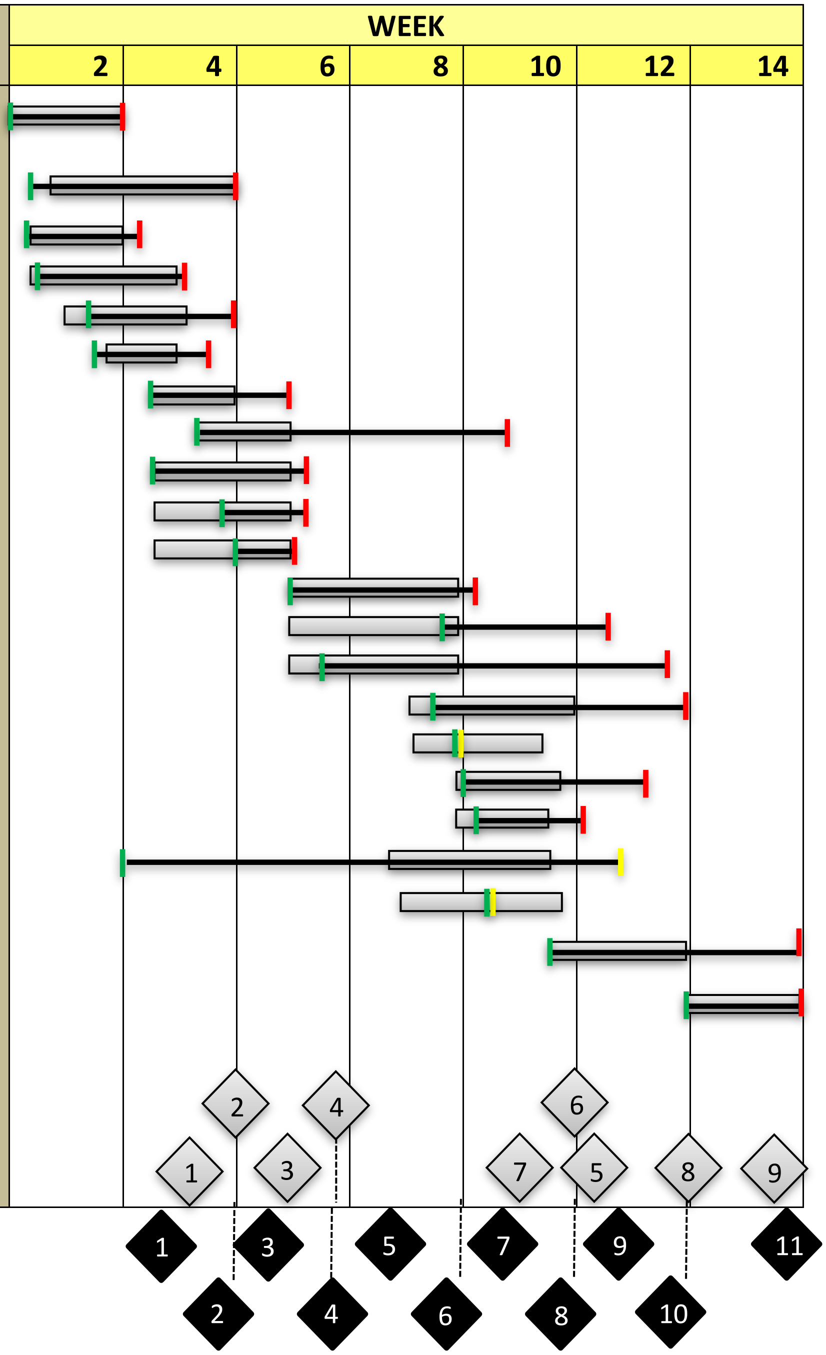


Figure 7.2.1: Gantt Charts displaying the Project Schedule

LEGEND	
Symbol/Abbreviation	**Representation**
MAH	Amr Mahmoud
MOM	Stefan Momic
NIS	Hamza Nisar

n	Milestone
n	Rolled Up Milestone
	Task
	Progress
	Task Start
	Task Completion
	Task Termination
A	Complete Project Planning including Conceptual Designs, Statement of Work and Scheduling
B	Complete the Design Proposal
C	Acquire solder and wires. Furthermore, acquire various sensors including ultrasonic sensors, infrared sensors, photoresistors and photocells for testing and experimentation.
D	Acquire building materials such as wood, aluminum sheets and the necessary hardware (motors, actuators, wheels, etc.).
E	Complete code for running the keypad and LCD along with the first version of the machine interface.
F	Complete the experimentation of the various electronic components and sensors.
G	Complete the detailed geometry, kinematics, and weight and moment of inertia estimations.
H	Complete the machine mobile base and install all the wheel actuators on the mobile base. Materials need to be acquired for use in the real prototype if the materials from the experimentation stage cannot be reused.
I	Provide a detailed analysis of currents and voltages needed. Furthermore all the power requirements are completed with their specifications finalized and the power supplies and cables necessary being acquired.
J	Complete all circuits for the actuators and sensors on the breadboard. Provide investigation of signal interference when all the drivers and sensors work together.
K	Complete the pseudo code for the final program.
L	Complete final machine code and download onto the PIC board.
M	Complete construction of the arm and the retraction actuator.

N	Assemble and debug all circuits on the solder board.
O	Complete the entire machine platform and mechanisms.
P	Complete construction of own microcontroller board.
Q	Complete all the wiring to connect all the sensors and actuators to the microcontroller board.
R	Complete the fabrication and testing of solder boards.
S	Complete implementation and code for Real Time Date/Time Logs, Permanent Logs and PC Interface.
T	Complete the circuits and microcontroller code for the Remote Operation. Assemble and test its operation.
U	Complete system testing, debugging and any last minute fabrication necessary to complete the robot. Note that if the Extra Design Features are not operation by Week 11 and the rest of the system is not fully operator the effort for the Bonus Feature implementation shall terminate.
V	Prepare presentation for Public Demonstration and ensure that the robot operates accurately.
Milestone #1	Complete sensor experimentation and decision making process on the sensors to be used. Finalize the project design in preparation of project construction.
Milestone #2	Complete and submit the Design Proposal for Review. Complete construction of the frame, and actuator circuits.
Milestone #3	Week 5 Evaluation: The analysis and/or acquisition of each subsystem are complete with the hardware identification and acquisition for each subsystem being complete.
Milestone #4	Ensure that all the materials needed for the construction of the arm and all the circuit elements have been acquired and implementation/construction has started.
Milestone #5	Have each subsystem complete and ready for subsystem debugging to ensure complete subsystem functionality come the Week 8 Evaluation.
Milestone #6	Week 8 Evaluation: Have each subsystem completely functional.
Milestone #7	Have the solder boards complete with all the actuators and sensors connected to the microcontroller. Furthermore, the machine platform is complete and the Extra Design Features have begun implementation.
Milestone #8	Week 10 Evaluation: The entire machine platform with all the actuators is complete and the Extra Design Features have been implemented. The integration of all the subsystems is complete but could still suffer from some defects or deficiencies.

Milestone #9	The Extra Design Features are fully functional and the machine is operating near full functionality.
Milestone #10	Week 12 Evaluation: Full functionality of the machine is achieved.
Milestone #11	The project has been Publicly Demonstrated and placed in the 97% percentile in terms of performance as defined in the Performance Evaluation of the RFP.
Rolled Up Milestone #1	This milestone was completed more or less on time.
Rolled Up Milestone #2	This milestone was completed on time.
Rolled Up Milestone #3	This milestone was completed on time, with the exception of the Electromechanical member only having to bottom of the frame complete.
Rolled Up Milestone #4	This milestone was completed ahead of schedule as all of the materials were obtained on time. Certain materials needed to be obtained later after design changes and/or due to damage to the original components.
Rolled Up Milestone #5	It was not until after Week 10 that this milestone was achieved. Although the Circuits and Microcontroller member were able to deliver a satisfactory performance during Week 8 Evaluation, the Electromechanical member did not have a complete frame until after the Week 10 Evaluation.
Rolled Up Milestone #6	It was competed that "this would be great for week 8" by our TA upon our presentation for the Week 10 Evaluation. Although we had some integration by Week 10, for the most part integration has not started until after Week 10. As a result, the completion of Milestone #6 is pushed back until Week 10.
Rolled Up Milestone #7	The solder boards were complete and the actuators that were located on the robot body were complete at this time. The machine platform was complete with the exception of the arm and the balconies for the ultrasonic sensors. The Extra Design Features have begun implementation, particularly the Real-time Date/Time Display.
Rolled Up Milestone #8	What was needed to be completed for Week 10 Evaluation was not complete until Week 12. For the evaluation the Circuits and Microcontroller member were able to present the H-Bridge IC Chip working with the yellow motors in the project kit and the IR Sensor circuits configured with the microcontroller board (giving readings of which sensors were detecting an object). At this point, the arm was still not finalized, but for the most part the electromechanical subsystem was complete.

Rolled Up Milestone #9	Only the Real-time Date/Time Display was competed at this time. The other Extra Design Feature goals were abandoned in Week 9 after realization that they were not feasible.
Rolled Up Milestone #10	Was not accomplished.
Rolled Up Milestone #11	Was not accomplished.

Table 7.2.3: Gantt Charts Legend.

7.3. Weekly Schedules

It was expected for the members to maintain a weekly to do list in their Engineering Design Handbooks. A summary of each member's individual to do list, with dates of completion and justifications for incompletion of tasks are presented in the subsections below.

7.3.1. Electromechanical Member

Week	Tasks
1 (January 4 -10, 2016)	• Attain the textbook to read the required chapters for the following week • Meet with team to discuss individual and collective priorities • Attend technical lectures and perform background research on various electromechanical methods • Extensively read the RFP, carefully outlining any questions or clarifications needed
2 (January 11-17, 2016)	• Attend the first lab and complete machine shop training and familiarizing with power tools, drills, and machinery, with an emphasis on safety • Purchase the design kit and start project planning
3 (January 18-24, 2016)	• Mostly experimentation and rapid prototyping of different bases with different wooden pieces. This is a first experience so I wanted to get as familiar as possible before beginning the construction of the robot • Began work on the electromechanical section of the proposal, and compiling a list of possible items and materials needed for the design
4 (January 25-31, 2016)	• Proposal: completed the Executive Summary, the introduction for the Technical Body alongside drawings and sketches of the different areas of functionality of the machine, wrote an AHP script in MATLAB which was easy to use but was completely

	neglected by the Circuits member since there was no actual contribution to the document. This script would have saved the circuits member hours of unnecessary work as it calculates all required parameters by asking the user questions regarding each solution. Also wrote a 3rd conceptual design entry, providing details on specific arrangement of sensors, in addition to the electromechanical subsection. • Produced a tentative list of motors and materials required for the electromechanical subsystem. Since it was a first exposure to a largely practical field, it was the hope that with future iterations, a better and deeper understanding of different mechanisms and parts may be realized. • The complete mobile base was constructed on Jan 29, with 5V DC yellow motors and accompanying tires, aluminum angles purchased from metal supermarkets, and caster wheels
5 (February 1-7, 2016)	• The first individual subsystem evaluation took place this week, and with the help of the circuits member, the base was able to be tested. It was noticed that base was slightly uneven, and so adjustments were made so that each wheel may be completely in contact with the ground • Also, when the motors were powered using batteries from the circuits member, it was noticed that the base would not go completely straight, but turns about 15˚ from the initial position • Long blocks of wood were salvaged from the machine shop so that it would be possible to produce a more accurate design of the arm retraction system. Still debating over a lead-screw linear actuator or a rack-and-pinion setup. Neither of which I've worked with before, so more research was done to understand which was feasible.
6 (February 8-14, 2016)	• Corner angles are installed onto the base in an effort to provide support for a suspended platform housing the arm retraction mechanism • From the Week 5 evaluation, it was unclear whether the yellow dc motors should be changed to the cylindrical gearmotors or proceeds as is. A lot of thought was put into this, with discussions with the lab technician, and fellow students
7 (Reading Week)	• Both lab sessions that were open were attended for their entire duration. All materials for the basic electromechanical subsystem had been acquired, including the wood for the arms, a wooden platform, the base and the corner angles. • A large amount of time was spent investigating whether using a lead-screw as a linear retractor was a viable option. Most other electromechanical members were also at this stage, with some not even completed their base. As such, I discussed with fellow members and the lab technician, and bought all the necessary materials for a lead-screw setup in anticipation for the week 8 evaluation. A 6V, 100rpm motor was selected, recalling that a

	good balance between motor torque and rotational speed minimizes sacrifices in rotational speed and by extension, allow for a faster linear actuation. • However, when the lead-screw setup was built, it was apparent that the speed of the retraction would be much slower than expected, approaching 30 seconds for a one way extension of a small wooden block coupled to the screw. • By the end of the week, a gear kit was ordered which included rack gears. It was decided after much consideration that a rack-and-pinion implementation would be the best suited. However, for the week 8 evaluation, there would be a detriment to both electromechanical member and the circuits member, both of which was highly regretted.
8 (February 22- 28, 2016)	• The lead-screw setup was demonstrated to the TA, and the lack of progress for the construction of the arm was explained to the TA in the hopes that he would understand my predicament of a failed idea. • The rack gears arrived later in the week and a spur gear was quickly press-fitted onto the motor and tested with the rack gear. 1 revolution of the motor would translate into 2.5” of linear distance, which was a great relief to the electromechanical member who spent all week researching ways to correctly implement rack-and-pinion into similar designs. • It was decided that the base needed to be remade with different motors, wheels, and casters since the circuits member argued there was no space for shaft encoders which were previously intended. It was repeatedly explained to the circuits member that the currently used single-shaft yellow dc motors could easily be replaced with double shaft motors and by drilling two additional holes in the frame of the base, the shafts could be extended to allow for optical encoders. However, the circuits member refuse to accept this design and demanded a completely new design of the base, to which the electromechanical member had no choice but to comply. It should be noted that at this stage, the circuits member continuously stressed that he prefers “the robot to work. I don’t care about the budget.” It was expected that the base he was expecting from the electromechanical would result in $20 additional costs. Again, the electromechanical had no choice but to comply.
9 to 11 (February 29- March 20)	• In week 9, the base was completely rebuilt with new purchased motors, wheels, caster wheels, motor hubs, M3 screws (purchased from a hobbyist store that was needed to be quickly located in order to finish the base) which had to be searched for to find the closest size that would make the base as even as possible. In addition, wood was acquired through the machine shop for the arm’s platform. It was at the end of this week that the circuits member inspected the base and demanded for it to

be rebuilt for a *third* time since he suspected the corner angles would “shear”. There was no such problem but the electromechanical member had no choice but to comply because it was the circuit member’s repeated argument that he “had done this before in gr9 tech”.

- The electromechanical member (me) and the circuits member were driven home on the Friday of week 9 by the circuit member’s mother, along with all the Design Kit and Project kit items. The robot along with the Design Kit was dropped off at my house and I spent all evening and the following morning correcting the “shearing problem” as referred to by the circuits member. Shortly after, the base was rebuilt and the electromechanical member and the microcontroller member arrived at the circuit member’s house to prepare for the Week 10 evaluation. The delay in remaking the base for a third time was such that the arm was unable to be built and installed in time for the Week 10 evaluation. However, the correct wood needed to attach to the robot’s body was previously acquired and was given to the circuits member for him to mount his IR sensors. A 15cm x 15cm Aluminum was used for the base, to give the circuits member more space for his electrical components.
- As a result of the delay in making the electromechanical subsystem, there was a lack of progress to show for the Week 10 evaluation. The electromechanical member came to the Tuesday and Thursday lab sessions, effectively spending more than double the amount of time to finish the subsystem. The clean and efficient design of the subsystem, especially the screwed rack gears onto the basswood, was complimented by the lab technician whose advice and guidance were critical factors in my success.
- Week 11 and the weekend prior to Week 12 were spent collaborating with the other members to quickly finish integration in the hopes of eventually testing. On March 18 there was a dispute between the circuits and the electromechanical team member in reference to where the emergency stop button should be placed. The circuits’ member insisted that the electromechanical member drill 3 holes into the wooden platform, 2 for his wires and 1 to fit the emergency stop. An alternative proposed by the electromechanical member was to attach it to the top of one of the corner angles since it would be easier for the electromechanical member to construct, saving time that could be better spent elsewhere but the circuits member insisted, saying that “it looks nicer in the middle.”
- The shaft encoders were built during this week. The electromechanical member had a successful idea of using a lathe to drill holes into the centre of the wheels to make space for a tap to cut threads inside of it. This allows for a 6-32 threaded rod to

	be inserted and removed much easily, and the encoders to be secured. This was a great benefit to all members because the shaft encoders were not permanent and could be removed to be calibrated, if necessary.
12 (March 21-27)	• Week 12 was spent trying to make the robot go as straight as possible. The circuits member was a huge help, and with his assistance, the rate of progress might have been much slower. The encoders were also fixed and ensured to measure distance as accurately as possible • The Week 12 evaluation was completed, and the TA was fine with allowing our group to proceed to the demonstration. A lot more functionality was demonstrated at this evaluation than ever before, restoring faith in the electromechanical member's ability to deliver despite delays and failures.
13 (March 28-April 3)	• The system debugging and calibration stage continued, and some modifications were performed to reduce the total budget cost to below the required amount of $230.
14 (April 4-8, 2016)	• This was the week of the project demonstration, and a lot of effort was put in by all 3 members, but an emphasis for the circuits member since a lot of days were spent at his home with the robot. The robot was disqualified, but the presentation to the TA went well, and the TA was impressed by the brochures designed and printed by the circuits member in an attempt to provide a good and lasting impression of our design, Detex.

7.3.2. Microcontroller Member

Week1: Attend all lectures and ensure that I get a firm understanding of the Microcontroller's responsibilities. Read chapters 1 to 3 to gain background information about the project.

Week2: Pay for the design and project kit and receive the microcontroller board which will be used for the rest of the project. Read chapters 4 and 7 to learn more about the project details and how they pertain to my system. Begin planning and sorting out the workload.

Week3: Start writing proposal draft. Work on the microcontroller board and try to perform some of the exercises on the PML website.

Week4: Finish design proposal. Begin Writing up the entire pseudo code for the final product and slowly begin programming it.

Week5: Finish all exercises on the PML and begin circuits testing and implementation with the circuit member. Week 5 deliverables which includes the completed pseudo code for the final draft.

7.3.3. Circuits Member

<table>
<tr><th>Week</th><th>Tasks</th><th>Date of Completion</th></tr>
<tr><td rowspan="4">1 (January 4 -10, 2016)</td><td>Preliminary Group Meeting to Discuss Goals</td><td>January 7, 2016</td></tr>
<tr><td>Read Chapters 1 to 3</td><td>January 10, 2016</td></tr>
<tr><td>Get money for Design and Project Kits</td><td>January 10, 2016</td></tr>
<tr><td>Log into University/Design Portals</td><td>January 6, 2016</td></tr>
<tr><td colspan="3">The highlight of week 1 was the preliminary group meeting in which a team goal was created. The team goal that was presented with ensuring qualification in the competition and attempt to hit the following bonuses: elegancy & safety, accuracy, real time date/time logs, permanent logs, remote operation and PC Interface. During this meeting how to make decisions, communication review times, meetings and a plan for resolving conflict were discussed. The notes of this are presented in Stefan Momic's Engineering Design Handbook, Volume 1, page 117.
Additionally, preliminary ideas of the design were brainstormed during this meeting.</td></tr>
<tr><td rowspan="4">2 (January 11-17, 2016)</td><td>Attend Lab #1 to complete the initial checkout, purchase the design and project kits and personality dimension surveys.</td><td>January 11, 2016</td></tr>
<tr><td>Review Chapter 4, 5, and 8</td><td>January 16, 2016</td></tr>
<tr><td>Start Project Panning – discuss the Statement of Work</td><td>January 17, 2016</td></tr>
<tr><td>Exploration (survey, background, brainstorm)</td><td>January 27, 2016</td></tr>
<tr><td colspan="3">The major part of week 2 was the beginning of writing the Project Proposal, when the Statement of Work and project planning was discussed. Formalized exploration was not done until closer to the Project Proposal deadline, although independent exploration was done prior to being individual work on subsystem. This exploration however was more so with what kind of sensors were best to be used, rather than big picture design features.</td></tr>
<tr><td rowspan="2">3 (January 18-24, 2016)</td><td>Start experimentation (work with the Driver Board and build simple circuits on the protoboard)</td><td>January 22, 2016</td></tr>
<tr><td>Write up proposal draft (includes a complete list of tasks and division of duties; Gantt Charts and PERT analysis)</td><td>January 26, 2016</td></tr>
<tr><td colspan="3">The major component of week 2 was the beginning of the experimentation phase. In terms of experimentation it was a rather slow week due to the lack of experience and materials bought.</td></tr>
</table>

Work was done using the Driver Board by going through the various activities located at PML4all. This week was also big on the Design Proposal with the Circuits member completing all the Gantt Charts, PERT analyses and AHP Decision Making matrices.		
4 (January 25-31, 2016)	Finish Design Proposal	January 28, 2016
	Complete circuit for actuator and sensors on protoboard	January 29, 2016
	Provide Power Specifications and signal interference investigation.	January 29, 2016
	Provide detailed analysis of currents and voltages; and finalized power requirements.	January 29, 2016
	Acquire power supplies and cables.	January 29, 2016
Most of the beginning of week 4 was dedicated to the design proposal with the exception of Monday's lab in which the IR Sensor circuits were experimented with. The experimentation done with the IR Sensors was regarding the optimal resistance value of the resistor in series with the IR Receiver. The optimum resistor was selected to be the one that had the greatest difference in voltage difference values between the readings done at ambient light, in front of white tape and in front of black tape. Determining the proximity sensor is addressed in greater detail in Appendix B. After submission of the Design Proposal on Thursday, the focus was preparing for Week 5 evaluation. An open H-Bridge, acquired Ultrasonic Sensors and the IR Sensor circuits were demonstrated for the evaluation. Batteries and wires were also purchased.		
5 (February 1-7, 2016)	Week 5 Evaluation	February 1, 2016
	Work towards subsystem completion.	February 12, 2016
	Assemble all circuits together in preparation for circuit implementation on a solder board.	February 12, 2016
The main highlight of week 5 is that of week 5 evaluation which went rather smoothly. After the evaluation, the Circuit member worked towards developing his subsystem for Week 8 evaluation. Primarily the IR Sensor circuits were finalized and prepared for soldering. The member also attempted to collaborate with the Microcontroller member to get the Ultrasonic sensors to work and purchased the Optical Encoders. Work on the open H-Bridges was not done due to the fact that the finalized motor selections were not known. However, the member did research on versatile H-Bridge ICs that could be used for the task irrespective of the final motor selection.		
6 (February 8-14, 2016)	Continuation of Week 5 To-Dos.	
With two midterms this week, and the NBA All-Star weekend coming up (of which I was fortunate enough to attend the Rising Stars Challenge), AER201 was put off except for work during the Monday Lab session (the Friday one was skipped due to the All-Star Game). Lost time was made up for during Reading Week next week.		
7 (Reading Week)	Solder IR Sensor Circuits	February 20,2016
	Obtain Ultrasonic Sensors (all 3)	February16, 2016
	Obtain Shift Encoders	February 16, 2016

	Solder Open H-Bridges	April 1, 2016

Much of Reading week was spent soldering the IR Sensor circuits on a 3cm solder board width so they could fit on the arm. During this time the ultrasonic sensors and shift encoders were also obtained. Circuitry for these was not needed since they directly connected into the microcontroller board. Ideally the open H-Bridge would have been completed at this time, but it was not known what motors we were using yet and the Circuits member did not want to put work into a circuit that could not be used. The making of the H-Bridge was delayed for the end of System Integration because the Circuits member prioritized getting the system to be functional and used an IC H-Bridge for testing, compromising marks for his own subsystem completion for the benefit of the team.

8 (February 22- 28, 2016)	Get the IR Sensors integrated with the microcontroller.	February 29, 2016
	Get the H-Bridge IC working with the Microcontroller	February 29, 2016

The Microcontroller and Circuits member were able to get the IR Sensors communicating with the microcontroller and were able to display which of the IR Sensors were detecting white. Additionally, they were able to get the H-Bridge to control the direction of motor rotation. This H-Bridge IC was determined to be used for the drive motors, and an Open H-Bridge was decided to be used for the Arm Motor.

9 to 11 (February 29- March 20)	Complete the arm (mount IR Sensors)	March 18, 2016
	Complete H-Bridge IC Circuits for Driving (solder to solder board)	March 8, 2016
	Make an Open H-Bridge for the Arm	March 18, 2016 (temporary)
	Complete Power Requirements and provide the power circuitry necessary to power the microcontroller and all the necessary circuits	March 18, 2016
	Obtain and install an emergency stop button	March 18, 2016
	Complete the IR Sensors on the body	March 6, 2016
	Interface and mount Ultrasonic Sensors	March 25, 2016
	Interface and mount Optical Encoders	March 25, 2016
	Mount Microcontroller board and connect all the jumper wires to the corresponding ports.	March 15, 2016

Weeks 9 to 11 were the main part of the project. A separate weekly to do list was not created for this time period with the deliverable goal being the Week 12 Evaluation. It was decided to present whatever was already complete for the Week 10 Progress Evaluation.

For the Week 10 Evaluation (March 7, 2016) only the IR Sensors were demonstrated and mounted to the robot's side. Hamza was busy building the base at the time, so this part was done entirely by the Circuits and Microcontroller members. It was them who obtained one of the 24” pieces of basswood and mounted the IR Sensors using push pins. The 24” piece of basswood was not even cut to size for the Evaluation. Additionally, the Microcontroller and

Circuits member were able to demonstrate the H-Bridge IC Circuit working through communication with the microcontroller. Soldering of the H-Bridge begun that lab section but it was not completed in time for the evaluation.

By Week 12, the base was finally complete and the other 4 IR Sensors which were completed for Week 8 were mounted to the arm and interfaced with the microcontroller. The ultrasonic sensors and encoders were also finally mounted and interfaced (the interfacing was accomplished during week 10, but their mounting on the robot was the rate determining step). All the other parts of the robot including the microcontroller mounts and an emergency stop button were installed in time for the Week 12 Evaluation. Unfortunately, due to the delay in completing the base, not much testing was able to be accomplished prior to Week 12, with the first real run taking place during the Week 12 Evaluation lab. It was Week 12 where we really started system calibration/configuration in an attempt to get it to full functionality. It is believed that if we had 2 more weeks the robot could have qualified in the competition. However, we were limited in time due to the inability of the Electromechanical member to deliver a completed base come Week 8 and thus taking away 2 weeks of potential trial runs and system debugging. Although it was possible to potentially work on getting the robot to drive straight without the competition of the arm, this was not possible due to the fact that Hamza kept the robot with himself because he needed to "level it" and install his other parts. During this time the sensors were all individually calibrated and tested and they showed functionality. However, factors such as driving straight were not able to be tested until the base was fully completed.

12 (March 21-27)	Make open H-Bridge for the Arm	April 1, 2016
	Run tests and debug	April 4, 2016

It was hoped for the open H-Bridge to be made on the protoboard and tested with the microcontroller during the optional labs taking place on the Friday's of week 9 and 10. However, Amr was unable to attend these labs due to the need to babysit a Jamaican family near Orangeville. During this time he took the microcontroller board and worked on his code. It made no sense to start soldering the H-Bridge during this time prior to checking if it actually worked. During week 11, system functionality was prioritized over making and testing an open H-Bridge. Ultimately, the Circuits member sacrificed his own subsystem competition for the benefit of a closer to functional robot in attempt to make it to the Project Demonstration.

The main testing that was done during this time was attempting to make the robot go straight. Debugging of this was done primarily by the Microcontroller member by playing around with the encoder readings and predetermining a PWM signal. During this time the Circuits member helped the Electromechanical member straighten the wheels and encoders which were poorly mounted in an attempt to make the robot go straighter. Other factors such as redistributing the weight were also considered in an attempt to make the robot go straight. Additionally, when the IR Sensors were mounted all in parallel the threshold voltage reading of an detected barrel was determined to be 0.5V. Knowing this new threshold value, the Microcontroller was informed to adjust the code for the logic to be that white is detected when a voltage reading is greater than 0.5V.

13 (March 28-April 3)	Finish Robot (run tests, calibrated, configure)	April 4, 2016
	Make a Brochure for the Demonstration	April 2, 2016

	Ensure the Cost of the Robot was within budget	April 1, 2016
An Excel spreadsheet was created to ensure that the cost of the robot was less than $230CAD. Nominal prices of the components used were obtained by searching for the prices of the parts used online. Additionally it was noticed that we were still slightly over budget. In an attempt to reduce our cost, we actually made design modifications to our robot with the design for a cheaper price in consideration (rather than making up the numbers like most groups bragged about doing). Realizing that each rack was $1.60, we took 2 of them off from the arm since they were not actually ever used. Additionally, we swapped out the 12V motor for a 6V motor for it's cheaper price and eliminated one of the motor brackets (which cost $4.74) and opted to use hot glue instead. Once the cost of the robot was obtained, a brochure was created, as evidenced by the heavy circuit component in terms of Project Concept. As before, much of the testing was done on making the robot go straight due to the desire of qualification. Since you only need to detect the barrels and avoid hitting anything during the trial to qualify, the ultrasonic sensors for detection and going straight were the priorities. Calibration of distance readings, along with determining the level of water was put on the side burner for the time being, with the focus being on driving straight. The IR Sensors for determining the water level were fully functional since Week 8 as evidenced by a LED light turning on when white tape was detected. The logic for this was not implemented in the microcontroller however due to the priority of driving straight. By the time of the Public Demonstration we managed to get the robot to go reasonably straight and it was able to stop and turn on a LED light upon detection of a barrel.		
14 (April 4-8, 2016)	Project Demonstration Day	April 4, 2016
	Complete Final Report	April 8, 2016
The project culminated in the Project Demonstration. Unfortunately, the team was unable to qualify. Ironically, the machine failed in ways that were never seen before during project testing beforehand. Previously, our machine would fail because it would go off course crashing into the barrels. Here however, it was able to go straight and detect a barrel, causing it to stop. Unfortunately, it was never able to restart again upon detecting the second barrel. It was hypothesized that the machine code was stuck in an interrupt service routine, as a result not causing the robot to restart again.		

8. Budget

DeteX © was created within the $230CDN allowable budget. In order to ensure the project was completed on budget, various design decisions and material selections needed to be modified closer to the end of the project deadline. A complete summary of the budget and the compromises is summarized in Table 8.1 below.

Item	Quantity	Cost	Total	Source
Microcontroller				
PIC DevBugger Development Board	1	$50.00	$50.00	Project Kit
LCD & Keypad	1	$6.00	$6.00	Project Kit
RTC Chip & Battery	1	$5.00	$5.00	Project Kit
Circuits				
HC-SR04 Ultrasonic Sensor	3	$2.78	$8.34	Gear Best
Infrared Sensors (LTR-4280 & LTR-3208)	8	$0.80	$6.40	Amazon.com
L293 H-Bridge	1	$3.00	$3.00	Digi-Key
5V Voltage Regulator	1	$0.26	$0.26	Digi-Key
Heat Sinks	5	$0.45	$2.25	Digi-Key
TIP147 Transistor	2	$0.87	$1.74	Digi-Key
TIP142 Transistor	2	$1.07	$2.14	Digi-Key
Battery Holders (8 & 4 Packs)	1	$4.00	$4.00	Home Hardware
Optical Encoders	2	$6.33	$12.66	AliExpress
InfiniCell AA Batteries	12	$0.31	$3.71	The Source
LEDs	8	$0.12	$0.96	Creatron
10k Potentiometer	8	$0.35	$2.80	A1 Electronics
L741 OP-AMP	8	$0.90	$7.20	Creatron
Jumper Wire Set (130 pack)	75 out of 130 wires	$6.99	$3.15	Amazon.ca
Resistors (100Ω, 330Ω, 1kΩ, 41kΩ)	28	$0.02	$0.56	Project Kit/Creatron
PIC Ports	20	$0.15	$3.00	Home Hardware
PIC Power Adaptor	1	$2.50	$2.50	Home Hardware
Off Switch	1	$0.50	$0.50	Project Kit
Solder Boards	2.5	$0.38	$0.95	AliExpress
Wires, Solder, Electrical Tape and Zip Ties	1	$4.00	$4.00	Robot Shop, Wal-Mart
Electromechanical				
6V Zheng DC Motors	3	$6.50	$19.50	AliBaba
Racks	5	$1.60	$8.00	Vex Robotics
Spur Gear	1	$1.20	$1.20	Vex Robotics
24" Basswood (1/4" x 3")	1.75	$1.68	$2.94	Balsa Wood
15.75" Aluminum Angle	4	$1.41	$5.64	Metal Supermarkets
11.875" Aluminum Angle	3	$1.06	$3.18	Metal Supermarkets
1/16 x 3/4 x 48" Aluminum Angle	1	$5.16	$5.16	Metals Debot
25D Aluminum Gear Motor Brackets	2	$4.74	$9.48	Canada Robotix

4mm Aluminum Motor Hubs	2	$4.44	$8.88	Canada Robotix
Red Wheel	2	$6.34	$12.68	Canada Robotix
2" Rigid Caster Wheels	2	$2.98	$5.96	Home Hardware
Nylon Standoffs (2 packs – 6mm, 10mm)	1	$2.90	$2.90	Creatron
1" & 1-1/2" Corner Brace	2	$1.30	$2.60	Home Hardware
1/2" x 7" x 11" Plywood	1	$3.00	$3.00	Home Hardware
Pinewood Block	1	$1.20	$1.20	Home Hardware
1' 6-32 Threaded Rod	1	$0.32	$0.32	Home Hardware
15x15cm Aluminum Sheet	1	$1.00	$1.00	Machine Shop
Screws and Nuts (Machine/Regular/Wood)	1	$5.00	$5.00	Home Hardware
TOTAL				**$229.77**

Table 8.1: Detailed Budget Breakdown

It is not coincidental that the budget is so close to being at the upper limit as certain design modifications were made during Week 13 to ensure the budget constraint was met. A ballpark budget estimate was kept throughout the term and at no point did it seem like we weren't able to meet the budget. As a result, only a detailed budget report was needed at the end to ensure we were under budget, knowing that the budget could be met with slight changes if need be. This feat was practical due to our Flexibility in Construction Acceptance Criteria which we valued highly. A couple of the main design changes made to ensure we were under budget are outlined below:

- The batteries used were changed from Duracell to InfiniCell. This ultimately resulted in a reduced cost from $0.67/battery to $0.31/battery saving a total of $4.32.
- The motor used for the arm actuator was changed from the 12V Zheng motor provided in the Project Kit to a 6V Zheng Motor, saving a total of $2.50.
- The amount of racks used was reduced from 7 to 5 upon noticing that 2 were never actually in contact with the gear. Noticing that each rack was $1.60, this enabled savings of $3.20.
- Seeing that each Motor Bracket was $4.74, it was decided to eliminate the motor bracket used to hold the arm and hold it with glue instead. This in turn resulted in $4.74 of savings.

The bullets highlighted above ultimately allowed for $14.76 of savings and brought us under the $230 budget. At this point design modifications were stopped, and the testing for system functionality continued.

9. Conclusions

Above all else, Detex© is a barrel inspecter prototype. Following construction, countless iterations and calibrations were made to improve its functionality. Though it did not qualify for the competition, the work done in the 14 weeks prior has its merits. The initial stages heavily involved experimentation and familiarization with the existing materials, components, and technology. Such a strategy was a crucial first step. Following the initial design stages, the fabrication of the subsystems took place. All members did their best to stick to the agreed upon schedules, and for the most part did not encounter difficulty, except for some delays in the delivery of the electromechanical subsystem. Close to the end of the fabrication stages, subsystem integration took place. It was then that all members began to take a generalist approach to design, helping each other with various tasks. Special consideration should be made for the circuits member as he is to be commended for his diligence and commitment throughout the course of this project. Whether the failure of the project was due to the code, the internal circuitry, or an electromechanical defect, the experience gained from building the machine remains invaluable. It is the hope of the design team that this work will help in building their careers in design and interdisciplinary engineering.

Lastly, a major improvement for the machine would be its capacity to drive in a straight trajectory. Throughout most of the integration, debugging, and calibration stages, the solution to this problem was sought. This was a problem that involved each of the three subsystems. Some solutions that were tried were adjusting wheel and motor alignment, self-correction using PWM, keeping distance measurements using shaft encoders, and calibrating such that the weight distribution would have as little an effect as possible.

It is estimated that an additional time of 2 weeks would help the project succeed and qualify for the competition. Although this is not possible, it should be noted that the rate of progress towards the end of the time schedule was much greater than in the beginning. This is the result of the accumulation of experience and practical knowledge of the members.

10. Standard Operating Procedure

11. References

[6, 7] Calin, Dragos. "Types of Sensors for Target Detection and Tracking." *Into Robotics*. N.p., January 2014. Web. 27 January 2016 <http://www.intorobotics.com/types-sensors-target-detection-tracking/>.

[5] Lion Precision. "Eddy-Current Displacement Sensors." *Eddy-Current Sensors for Displacement and Position Sensing*. N.p., n.d. Web. 24 January 2016. <http://www.lionprecision.com/eddy-current-sensors/>.

[1, 1, 2] Harris, Tom. "How Robots Work." *HowStuffWorks? Science.* HowStuffWorks, n.d. Web. 3 April 2016. <http://science.howstuffworks.com/robot4.htm>.

[3, 4] Weber, Marc. "Where To? A History of Autonomous Vehicles." *CHM Blog Where to A History of Autonomous Vehicles Comments.* Computer History Museum, 2014. Web. 03 April 2016. <http://www.computerhistory.org/atchm/where-to-a-history-of-autonomous-vehicles/>.

[1] Calin, Dragos. "2 Simple Methods To Choose Motors For Wheel Drive Robots". *Into Robotics*. N.p., 2016. Web. 7 February 2016.

[1]"Torque and Efficiency Calculation | Simple Electric Motors", *Simplemotor.com*, 2016. [Online]. Available: http://simplemotor.com/torque-and-efficiency-calculation/. [Accessed: 08- Apr- 2016].

[2]"DC Motor Calculations: Calculating Torque | MICROMO", *Micromo.com*, 2016. [Online]. Available: http://www.micromo.com/technical-library/dc-motor-tutorials/motor-calculations. [Accessed: 08- Apr- 2016].

12. Bibliography

Elec Freaks. *Ultrasonic Ranging Module HC - SR04* (n.d.): 1-3. Web. 8 April 2016. <http://www.micropik.com/PDF/HCSR04.pdf>.

Spec No: Ds-50-92-0015. *IR Emitter and Detector* (n.d.): 1-4. LITEON Optoelectronics. Web. 8 April 2016. <http://optoelectronics.liteon.com/upload/download/DS-50-92-0015/S_110_LTE-4208.pdf>.

Spec No: Ds-50-92-0068. *IR Emitter and Detector* (n.d.): 1-4. LITEON Optoelectronics. Web. 8 April 2016. <http://optoelectronics.liteon.com/upload/download/DS-50-92-0068/S_110_R3208E.pdf>.

ST Microelectronics. "High-Voltage Darlington Transistors." *Electronic Devices on Discrete Components for Industrial and Power Engineering* Rev 5 (2008): 333-40. ST Microelectronics, 2010. Web. 8 April 2016. <http://www.st.com/web/en/resource/technical/document/datasheet/CD00000917.pdf>.

Texas Instruments Incorporated. *L293, L293D Quadruple Half-H Drivers (Rev. D)* (2016): n. pag. January 2016. Web. 8 April 2016. <http://www.ti.com/lit/ds/symlink/l293d.pdf>.

13. Tables

14. Figures

15. Appendices

15.1. Appendix A – fd s

15.2. Appendix B – Circuit Experimentation Results

<u>Determining the Proximity Sensor Needed for Barrel Detection</u>

For this experiment was conducted using the following circuit schematic.

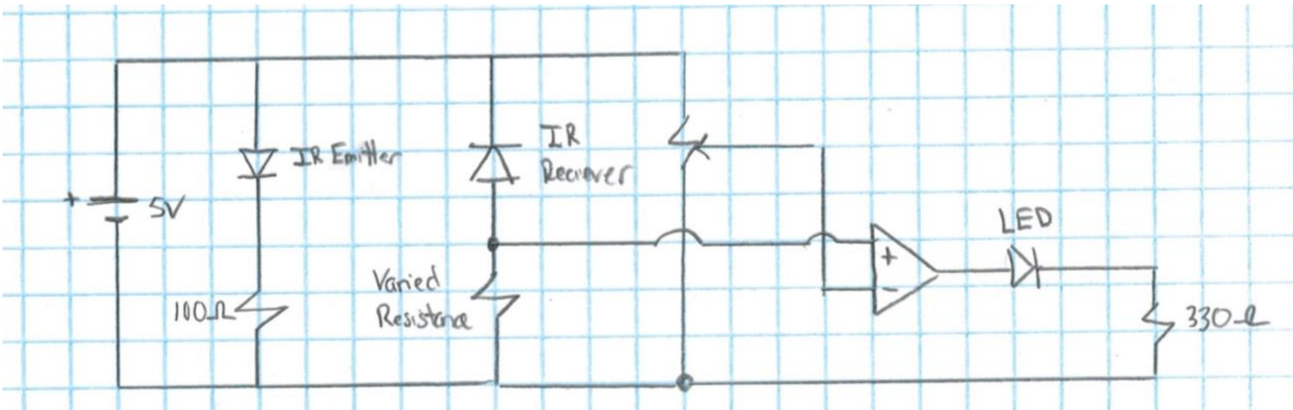


Figure B1: Schematic Diagram of Experimentation Method

The resistance in series with the IR Receiver was varied with the hopes of changing the sensitive range of the sensor. The results of the experiment are summarized in the table below.

Resistance Value in Series [kΩ]	Colour Detecting (at a 3cm distance)	Voltage Drop Across the Infrared Receiver [V]	LED Response
30	Ambient Light	5.50	OFF
	Black Barrel	1.40	ON
	White Tape	5.07	OFF
40	Ambient Light	4.10	OFF
	Black Barrel	2.05	ON
	White Tape	4.73	OFF
50	Ambient Light	4.06	OFF
	Black Barrel	0.35	ON
	White Tape	4.56	OFF
60	Ambient Light	4.76	OFF
	Black Barrel	0.26	ON
	White Tape	4.86	OFF

Table B1: Experimentation Results for IR Sensor Testing

As is evident from the table the voltage difference between the black barrel and ambient light is too small a range to confidently be used to detect the presence of a barrel. As a result it is evident that the infrared sensor is not a good option as a sensor used for barrel detection.

<u>Determining the Method for Water Level Indication</u>

Method	Colour Detection	Voltage Drop Across Receiver [V]	Difference between Black and White
Photo Resistor	Black Barrel	4.9	0.57
	White Tape	4.33	

IR Receiver (50kΩ resistor in series)	Black Barrel	4.56	4.21
	White Tape	0.35	

Table B2: Experimentation Results for determination of the Method for Water Level Measurement

It is evident that using an IR Receiver creates a greater voltage difference and hence will be easier to assign the threshold value for white tape detection during sensor calibration. As a result, an IR Receiver is the preferable method for determining the water level.

15.3. Appendix C – Robot Main Assembly Code

It should be noted that the code presented does not include code for the LCD, i2c assembly and header files.

```
    list p=16f877               ; list directive to define processor
    #include <p16f877.inc>      ; processor specific variable definitions
    __CONFIG _CP_OFF & _WDT_OFF & _BODEN_ON & _PWRTE_ON & _HS_OSC &
_WRT_ENABLE_ON & _CPD_OFF & _LVP_OFF

    #include <lcd.inc>                      ;Import LCD control functions from lcd.asm
        #include<rtc_macros.inc>
        Ones EQU 0X30 ;1's minutes
        Tens EQU 0X31 ;10's minutes
        Hundreds EQU 0X32 ;100's
        Stat1 EQU 0X33 ;1's minutes
        Distance1   EQU     0x34
        Distance2   EQU     0x35
        ADDR  EQU    0x36
        VALUE EQU    0x37
 Ones1 EQU 0X38 ;1's minutes
Tens1 EQU 0X39 ;10's minutes
Hundreds1 EQU 0X3A ;100's
 Ones2 EQU 0X3B ;1's minutes
Tens2 EQU 0X3C ;10's minutes
Hundreds2 EQU 0X3D ;100's
 l      EQU 0x3E
m       EQU 0x3F
n       EQU 0x40
o       EQU 0x41

Count   EQU 0x42
One EQU 0x43
barrel  EQU 0x44
```

```
	curr	EQU 0x45
	Temp30		EQU 0x46
	z  equ  0x47
  y	equ 0x48
	x  equ  0x49
  Temporarycount equ		0x4A
 Countr		equ 0x4B
	udata	0x25
	COUNTH		res 1	;const used in delay
	COUNTM		res	1	;const used in delay
	COUNTL		res	1	;const used in delay
	Table_Counter		res	1
	temp		res 1				; Variable used for storage\
	TEMP	res	1
	LASTPB		res	1
	tempb	res	1
	tempa	res	1
	barreldetected		res 1

columndetected		res 1
		lcd_tmp		res	1
lcd_d1	res	1
lcd_d2	res	1
com		res	1
dat		res	1

	;Declare constants for pin assignments (LCD on PORTD)

	cblock
	d1
	endc
#define		RS	PORTD,2
#define		E	PORTD,3

      org 0x00
	reset
	goto init
	org 0x04
	goto		IntSRV
	org		0x100
IntSRV
	;btfsc	INTCON,T0IF
	;goto	encoder
	;bcf		T1CON,TMR1ON
```

```
        bcf     PIR1,TMR1IF

        movlw   0xFF
        movwf       TMR1H
        movlw       0xB8
        movwf       TMR1L

        btfsc   PORTA,1
        incf Ones,f ;increase 1's count
        btfsc   PORTA,2
        incf    Ones1,f
        btfsc   PORTA,3
        incf    Ones2,f

encoder
        bcf INTCON,T0IF
MOVFW PORTB ; Move PortB value to the W register
 ; This ends mismatch conditions
 MOVWF TEMP ; Need to save the PortB reading.
 XORWF LASTPB ; XOR last PortB value with the new
 ; PortB value.

 BTFSC LASTPB,0 ; Did pin RB5 change
 CALL RB0_CHG ; RB0 changed and caused the interrupt
BTFSC LASTPB, 2 ; Did pin RB4 change
 GOTO RB2_CHG ; RB2 changed and caused the interrupt
goto    CLR_RBINTF

RB0_CHG
        incf    One
        RETURN
RB2_CHG
        incf    Count

CLR_RBINTF
        MOVFW TEMP ; Move the PortB read value to the
        MOVWF LASTPB ; register LASTPB
        movlw d'255'
        subwf  Count,w
        btfss   STATUS,Z
        goto    nextr
        incf    Countr,f
        clrf    Count
nextr
```

```
        movlw d'0'
        subwf  Countr,w
        btfss   STATUS,Z
        call CheckDistance
        call    checkencoder
continue
        goto    next
;------------------------------------------------------------------------

next
        retfie
checkencoder
        movfw One
        subwf  Count,w
        btfsc   STATUS,Z
        goto    gostraight
        btfsc   STATUS,C
        goto    turnleft
        goto    turnright
gostraight
        bsf PORTC,0
        bsf PORTC,1
        goto    continue
turnleft
        bcf     PORTC,0
        bsf     PORTC,1
        goto    continue
turnright
        bcf     PORTC,1
        bsf     PORTC,0
        goto    continue
;**************************************
; Display macro
;**************************************
Display macro Message
                local   loop_
                local   end_
                clrf    Table_Counter
                clrw
loop_   movf    Table_Counter,W
                call    Message
                xorlw   B'00000000' ;check WORK reg to see if 0 is returned
                btfsc   STATUS,Z
                        goto    end_
```

```
            call    WR_DATA
            incf    Table_Counter,F
            goto    loop_
end_
            endm
;-----------------------------
; Change Page
;-----------------------------
PageChange  macro  tableIndex
        movwf       temp                        ; Save the index
        movlw       HIGH   tableIndex
        movwf       PCLATH                              ; Store it for next jump
        movf        temp, W
        addlw       LOW         tableIndex      ; Compute its Offset
        btfsc       STATUS, C
        incf    PCLATH, F                   ; If in next, PCLATH = PCLATH + 1
        movwf       PCL                         ; Write the correct address to PC
        endm                        ; End of macro

;*************************************
; Initialize LCD
;*************************************
init
        ;Ports
      clrf    INTCON
      bsf     STATUS,RP0    ; select bank 1
         movlw0x06
         movwf          ADCON1                  ;ALL PORT A AND E IS ANALOG
         clrf     OPTION_REG
         movlwb'11111110'
         movwf    TRISA
      movlw    b'11110111'    ; Set required keypad inputs
      movwf    TRISB
         movlw   b'00011000'
         movwf            TRISC
      clrf    TRISD        ; All port D is output
         clrf       TRISE
      bcf     STATUS,RP0    ; select bank 0
      clrf    PORTA
      clrf    PORTB
         movlw   b'10101000'   ; Forward mortion for engines locked in
         movwf            PORTC

      clrf    PORTD
```

```
clrf            PORTE

    ;variables
    movlwd'0'
    movwf           ADDR
    movwf           barrel
    clrf Ones
    clrf Tens
    clrf Hundreds
    clrf Ones1
    clrf      Countr
    clrf Tens1
    clrf Hundreds1
    clrf Ones2
    clrf Tens2
    clrf      Distance2
    clrf Hundreds2
    clrf TEMP
    clrf      Count
    clrf      One
    clrf      l
    clrf      m
    clrf      n
    clrf      o
    movlw b'11111111'
    movwf LASTPB
    ;Timer
clrf          T1CON
movlw         0xFF
movwf         TMR1H
movlw                       0xB8
              movwf                     TMR1L
clrf          INTCON
bsf           STATUS,RP0
clrf          PIE1
bcf                         STATUS,RP0
clrf          PIR1
movlw         0x20
movwf         T1CON ;Prescaller 1:4
bsf                                     INTCON,PEIE
              ;bsf                             INTCON,GIE
                                                            ;TMR1H:TMR1L = 0X0000
                                                            ;           Td = 200ns*(2^16-
TMR1H:TMR1L)*PS
```

```
		bcf					INTCON,GIE

		bsf					STATUS,RP0
		bsf					PIE1,TMR1IE
	bcf					STATUS,RP0

		bsf		T1CON,TMR1ON
;		bcf		STATUS, RP0		;Bank 0
		bsf		INTCON,T0IE
		bsf INTCON,GIE
		 bsf	PORTD,0
		 bsf	PORTE,2
		  call	i2c_common_setup
	    call	InitLCD
		;bsf	STATUS,RP0
		;clrf	TRISC
;*************************************
; Main code
;*************************************


Main
		bcf	STATUS,RP0
		bcf	STATUS,RP1
		call		Clear_Display
		call		ResetVariables
		movlw	0x00
		movwf	ADDR
		clrf		VALUE

Start
		Display		DeteX
	      clrf	PCLATH
		call Switch_Lines
		Display		StartMessageTwo
		clrf	PCLATH
		 bsf	PORTD,0
		 bsf	PORTE,2
PollStart
		btfss		PORTB,1				; Poll until key pressed
		goto		$-1
		swapf		PORTB, W			; Copy RB4-7 into W0-3
		andlw		h'0F'
		xorlw		b'00000000'			; Check if 1 was pressed
```

```
                btfss           STATUS,Z
                goto            PollStart
                call    Show_RTC
                call    Clear_Display
                 bsf    PORTD,0
                 bsf    PORTE,2
                rtc_resetAll    ;reset rtc
                rtc_set 0x00,   B'10000000'
                rtc_set 0x01,   B'00000000'             ; Minutes
                rtc_set 0x00,   B'00000000'             ; Seconds

Inspect

                movlw 0x80
                call    WR_INS
                Display         Inspecting
                clrf       PCLATH
                movlw d'7'
                subwf  barrel,w
                btfsc   STATUS,Z
                goto    EndInspection
                 bsf    PORTD,0
                 bsf    PORTE,2
                bsf INTCON,GIE
                call            StartEngine1

                call            Trig
                call    WaitEchoHigh
                call            Trig
                call    WaitEchoHigh
                call            CheckUltra

CheckDistance
                movlw d'255'
                subwf  Count,w
                btfsc STATUS,Z
                goto    EndInspection
                return
SelectBank3
                bcf     STATUS,RP0
        BSF     STATUS,RP1
        return
CheckStatusTall
                call    SelectBank3
```

```
            incf    ADDR,f
            movlw   d'36'
            movwf   VALUE
            call    WriteData

            movfw   Count
            movwf   Temporarycount
            call    BCD2
            call    SelectBank3
            incf    ADDR,F
            movfw   z
            movwf   VALUE
            call    WriteData

            call    SelectBank3
            incf    ADDR,F
            movfw   y
            movwf   VALUE
            call    WriteData

            call    SelectBank3
            incf    ADDR,F
            movfw   x
            movwf   VALUE
            call    WriteData

            incf    barrel,f
            clrf    Stat1
            btfsc   PORTA,5
            incf    Stat1,f
            btfsc   PORTA,7
            incf    Stat1,f
            btfsc   PORTA,0
            incf    Stat1,f
            btfsc   PORTC,3
            incf    Stat1,f
            goto    WriteStatus
WriteStatus
            movlw   d'2'
            subwf   Stat1,w
            btfsc   STATUS,Z
            goto    WriteFull
            movlw   d'1'
            btfsc   STATUS,Z
```

```
            goto    WriteHalf
            goto    WriteEmpty
WriteFull
            call    SelectBank3
            incf    ADDR,F
            movlw d'22'
            movwf VALUE
            call    WriteData
            call    SelectBank3
            incf    ADDR,F
            movlw d'90'
            movwf VALUE
            call    WriteData
            ;return
            goto    Inspect
WriteHalf
            call    SelectBank3
            incf    ADDR,F
            movlw d'24'
            movwf VALUE
            call    WriteData
            call    SelectBank3
            incf    ADDR,F
            movlw d'22'
            movwf VALUE
            call    WriteData
            goto    Inspect
WriteEmpty
            call    SelectBank3
            incf    ADDR,F
            movlw d'21'
            movwf VALUE
            call    WriteData
            call    SelectBank3
            incf    ADDR,F
            movlw d'90'
            movwf VALUE
            call    WriteData
            goto    Inspect
CheckStatusShort
            call    SelectBank3
            incf    ADDR,F
            movlw d'35'
            movwf VALUE
```

```
                call    WriteData

                movfw Count
                movwf Temporarycount
                call    BCD2
                call    SelectBank3
                incf    ADDR,F
                movfw z
                movwf VALUE
                call    WriteData

                call    SelectBank3
                incf    ADDR,F
                movfw y
                movwf VALUE
                call    WriteData

                call    SelectBank3
                incf    ADDR,F
                movfw x
                movwf VALUE
                call    WriteData

                incf    barrel,f
                clrf    Stat1
                btfsc   PORTA,4
                incf    Stat1,f
                btfsc   PORTA,6
                incf    Stat1,f
                btfsc   PORTA,0
                incf    Stat1,f
                btfsc   PORTC,4
                incf    Stat1,f
                goto    WriteStatus
EndInspection
                movlw 0x80
                call    WR_INS
                Display   ReturningHome
                clrf    PCLATH
                call            Reverse
                call            HalfS
                call            HalfS
                call            StopEngine1
                call            RetractArm
```

```
            movfw       One
            movwf Count
            clrf    One
            bsf INTCON,GIE
            call        Reverse
            call        ReturnHome
            BCF INTCON,GIE
            call        HalfS
            call        StopEngine1
            goto        StopInspection

CheckUltra
            movlw 0x09
            subwf  Ones,w
            btfsc   STATUS,Z
            goto    NOLED
            btfsc   STATUS,C
            goto    LEDON
            goto    NOLED
LEDON
        call    StartEngine1
        goto    Inspect
NOLED
        call    Trig
        call    WaitEchoHigh
        movlw 0x09
        subwf  Ones,w
        btfsc   STATUS,C
        goto    Inspect
        call    Trig
        call    WaitEchoHigh
        movlw 0x09
        subwf  Ones,w
        btfsc   STATUS,C
        goto    Inspect
        call    StopEngine1
        movlw 0x09
        subwf  Ones2,w
        btfsc   STATUS,Z
        goto    DodgeColumn
        btfsc   STATUS,C
        goto    DetectedBarrel3
        goto    DodgeColumn
BCD2
```

```
 clrf     x
 clrf     y
 clrf     z
Lala2_
        decfsz  Temporarycount,f
        goto    inc2_
        return
inc2_
        incf    x,f
        movlw 0x0A
        subwf x,w
        btfss STATUS,Z
        goto Lala2_
        clrf x
        incf y ;increase 10's count
        movlw 0x0A
        subwf y,w
        btfss STATUS,Z
        goto Lala2_
        clrf y
        incf z ;increase 100's count
        movlw 0x0A
        subwf z,w
        btfss STATUS,Z
        goto Lala2_
        clrf z
        goto    Lala2_
DetectedBarrel3
        bsf PORTB,3
        call    Trig
        call    WaitEchoHigh
          movlw       0x09
                subwf   Ones,w
                btfsc   STATUS,Z
                goto    NOLED1
                btfsc   STATUS,C
                goto    LEDON1
                goto    NOLED1

LEDON1
         call    Trig
         call    WaitEchoHigh
         movlw0x09
                subwf   Ones,w
```

```
                btfss   STATUS,C
                goto    NOLED1
        call    Trig
        call    WaitEchoHigh
        movlw0x09
                subwf   Ones,w
                btfss   STATUS,C
                goto    NOLED1
        call    StopEngine1
        call    Clear_Display
        bcf     PORTB,3
        Display Detected
        clrf    PCLATH
        bsf     PORTD,0
                bsf     PORTE,2
        call    HalfS
        call    Clear_Display
        movlw0x09
                subwf   Ones1,w
                btfsc   STATUS,Z
                goto    CheckStatusTall
                btfsc   STATUS,C
                goto    CheckStatusShort
                goto    CheckStatusTall
NOLED1
        call    StartEngine1
        goto    DetectedBarrel3

ResetVariables
        clrf Ones
        clrf Tens
        clrf Hundreds
        clrf Ones1
        clrf Tens1
        clrf Hundreds1
        clrf Ones2
        clrf Tens2
        clrf Hundreds2
        return
DodgeColumn
                call    Clear_Display
                call            StopEngine1
                call            RetractArm
                call            StartEngine1
```

```
                call            HalfS
                call            HalfS
                call            StopEngine1
                call            ExtendArm
                goto            Inspect
StopInspection
                bcf PORTC,0
                bcf PORTC,1
                call    Clear_Display
                Display         Finish1
                clrf      PCLATH
                call      Switch_Lines
                movlw "R"
                call    WR_DATA
                movlw "u"
                call    WR_DATA
                movlw "n"
                call    WR_DATA
                movlw " "
                call    WR_DATA
                movlw "t"
                call    WR_DATA
                movlw "i"
                call    WR_DATA
                movlw "m"
                call    WR_DATA
                movlw "e"
                call    WR_DATA
                movlw ":"
                call    WR_DATA
                movlw " "
                call    WR_DATA
                bsf     STATUS,RP0
                bcf     STATUS,RP1
                movlw b'00011000'
                movwf TRISC
                bcf     STATUS,RP0
                bcf     STATUS,RP1
                ;Get minute
                rtc_read        0x01            ;Read Address 0x01 from DS1307---min
                movfw 0x78
                call    WR_DATA
                movlw ":"
                call    WR_DATA
```

```
        ;Get seconds
        rtc_read        0x00            ;Read Address 0x00 from DS1307---seconds
        movfw 0x77
        call    WR_DATA
        movfw 0x78
        call    WR_DATA
        ;Display                Finish2
        ;clrf       PCLATH
        call    HalfS
        movlw     0x00
        movwf     ADDR
        movlw 0x01
        movwf     curr
PollStart2
        btfss           PORTB,1                         ; Poll until key pressed
        goto            $-1
        swapf           PORTB, W                ; Copy RB4-7 into W0-3
        andlw           h'0F'
        xorlw           b'00000000'             ; Check if 1 was pressed
        btfss           STATUS,Z
        goto            PollStart2
        call    Clear_Display
        call    SelectBank3
        movlw 0x00
        movwf ADDR
        bcf STATUS,RP0
        bcf STATUS,RP1
        movlw 0x30
        movwf Temp30
ShowInfo
        call    Clear_Display
        ;movlw0x80
        ;call   WR_INS

        ;Display        Barrel#
        movlw 'B'
        movwf 0x20
        call    WR_DATA
        movlw 'a'
        movwf 0x20
        call    WR_DATA
        movlw 'r'
        movwf 0x20
```

```
call	WR_DATA
movlw	'r'
movwf	0x20
call	WR_DATA
movlw	'e'
movwf	0x20
call	WR_DATA
movlw	'l'
movwf	0x20
call	WR_DATA
movlw	'#'
movwf	0x20
call	WR_DATA
 bsf	PORTD,0
 bsf	PORTE,2
;clrf	PCLATH
movlw	0x30
addwf	curr,w
movwf	0x20
call	WR_DATA

DisplayTypes
clrf	PCLATH
call	SelectBank3
incf	ADDR,f
call	ReadData
addwf	Temp30,w
movwf	0x20
call	WR_DATA
bsf PORTD,0
bsf PORTE,2
call	HalfS
call	Switch_Lines

DisplayDist
clrf	PCLATH
call	SelectBank3
incf	ADDR,f
call	ReadData
addwf	Temp30,w
movwf	0x20
call	WR_DATA

call	SelectBank3
```

```
        incf    ADDR,f
        call    ReadData
        addwf   Temp30,w
        movwf   0x20
        call    WR_DATA

        call    SelectBank3
        incf    ADDR,f
        call    ReadData
        addwf   Temp30,w
        movwf   0x20
        call    WR_DATA

         Display        Stat
        clrf    PCLATH
        call    SelectBank3
        incf    ADDR,f
        call    ReadData
        addwf   Temp30,w
        movwf   0x20
        call    WR_DATA
        call    SelectBank3
        incf    ADDR,f
        call    ReadData
        addwf   Temp30,w
        movwf   0x20
        call    WR_DATA

        bsf     PORTD,0
         bsf    PORTE,2
PollData
        bsf     PORTA,0
        btfss           PORTB,1                         ; Poll until key pressed
        goto            $-1
        swapf           PORTB, W                ; Copy RB4-7 into W0-3
        andlw           h'0F'
        xorlw           b'00000000'             ; Check if 1 was pressed
        btfsc           STATUS,Z
        goto            PreviousData
        swapf           PORTB, W                ; Copy RB4-7 into W0-3
        andlw           h'0F'
        clrf            PORTB
        xorlw           b'00000001'             ; Check if 2 was pressed
        btfsc           STATUS,Z
```

```
                goto            NextData
                goto            PollData
PreviousData
                bcf PORTA,0
                movlw d'1'
                subwf   curr,w
                btfsc   STATUS,Z
                goto    PollData
                decf    curr,f
                movlw   d'6'
                call    SelectBank3
                subwf   ADDR,f
                bcf STATUS,RP0
                BCF STATUS,RP1
                goto    ShowInfo
NextData
                bcf PORTA,0
                movfw   barrel
                subwf   curr,w
                btfsc   STATUS,Z
                goto    PollData
                incf    curr,f
                movlw   d'6'
                call    SelectBank3
                subwf   ADDR,f
                bcf STATUS,RP0
                BCF STATUS,RP1
                goto    ShowInfo

ExtendArm
                bsf     PORTE,2
                bcf     PORTD,0
                call    HalfS
                call    HalfS
                call    HalfS
                call    HalfS
                call    HalfS
                call    HalfS
                call    HalfS
                call    HalfS
                call    HalfS
                bsf PORTD,0
                return
RetractArm
```

```
                    bsf     PORTD,0
                    bcf     PORTE,2
                    call    HalfS
                    call    HalfS
                    call    HalfS
                    call    HalfS
                    call    HalfS
                    call    HalfS
                    call    HalfS
                    call    HalfS
                    call    HalfS
                    call    HalfS
                    bsf     PORTE,2
                    return
ReturnHome
                    BSF INTCON,GIE
                    call    Reverse
                    movfw Count
                    subwf  One,w
                    btfsS   STATUS,Z
                    goto    ReturnHome
                    return

ReadData
                    BCF STATUS, RP0 ; Bank0
                    BSF STATUS,RP1
                    MOVFW           ADDR
                    MOVWF           EEADR
                    BSF STATUS,RP0
                    BCF EECON1,EEPGD
                    BSF EECON1,RD
                    BCF STATUS,RP0
                    MOVFW           EEDATA
                    bcf STATUS,RP1
          RETURN
WriteData
 bcf      INTCON,GIE
 bcf      STATUS,RP0
 BSF      STATUS,RP1
 MOVFW          ADDR
MOVWF           EEADR
 MOVFW          VALUE
 MOVWF          EEDATA
 bsf      STATUS,RP0
```

```
 BCF     EECON1,EEPGD
 BSF     EECON1,WREN
 MOVLW 0x55 ;
 MOVWF EECON2 ; 55h must be written to EECON2
 MOVLW 0xAA ; to start write sequence
 MOVWF EECON2 ; Write AAh
 BSF EECON1,WR ; Set WR bit begin write
 BTFSC EECON1,WR
 GOTO $-1
 BCF     EECON1,WREN
 BCF     STATUS,RP0
 BCF     STATUS,RP1
 BSF INTCON, GIE ; Enable INTs
RETURN

Trig
 bcf      INTCON,GIE
   call    ResetVariables
   movlw   b'00000111'
   movwf   PORTE
   call delayL
   clrf PORTE
  bsf     PORTE,2
   return
 WaitEchoHigh
   btfss PORTA,3 ;Test for Echo input High and start timer
   goto WaitEchoHigh
   bsf     INTCON,GIE
WaitEchoLow
   btfsc PORTA,3  ;Wait for Echo input Low and stop timer
   goto WaitEchoLow
   bcf    INTCON,GIE
   return
StartEngine1
          movlwb'10101011'
          movwf           PORTC
          return
StopEngine1
          clrf      PORTC
          return
Reverse
          movlwb'01010111'
           movwf           PORTC
          return
```

```
Show_RTC
            ;clear LCD screen
            movlw b'10000000'
            call    WR_INS

            ;Get year
            movlw "2"                          ;First line shows 20**/**/**
            call    WR_DATA
            movlw "0"
            call    WR_DATA
            rtc_read        0x06               ;Read Address 0x06 from DS1307---year
            movfw 0x77
            call    WR_DATA
            movfw 0x78
            call    WR_DATA

            movlw "/"
            call    WR_DATA

            ;Get month
            rtc_read        0x05               ;Read Address 0x05 from DS1307---month
            movfw 0x77
            call    WR_DATA
            movfw 0x78
            call    WR_DATA

            movlw "/"
            call    WR_DATA

            ;Get day
            rtc_read        0x04               ;Read Address 0x04 from DS1307---day
            movfw 0x77
            call    WR_DATA
            movfw 0x78
            call    WR_DATA

            movlw B'11000000'                  ;Next line displays (hour):(min):(sec) **:**:**
            call    WR_INS

            ;Get hour
            rtc_read        0x02               ;Read Address 0x02 from DS1307---hour
            movfw 0x77
            call    WR_DATA
            movfw 0x78
```

```
		call	WR_DATA
		movlw			":"
		call	WR_DATA

		;Get minute
		rtc_read	0x01		;Read Address 0x01 from DS1307---min
		movfw 0x77
		call	WR_DATA
		movfw 0x78
		call	WR_DATA
		movlw ":"
		call	WR_DATA

		;Get seconds
		rtc_read	0x00		;Read Address 0x00 from DS1307---seconds
		movfw 0x77
		call	WR_DATA
		movfw 0x78
		call	WR_DATA

		call	HalfS
		call	HalfS
		call	HalfS
		call	HalfS;Delay for exactly one seconds and read DS1307 again
		return
Delay10us
; Delay = 1e-005 seconds
; Clock frequency = 10 MHz

; Actual delay = 1e-005 seconds = 25 cycles
; Error = 0 %


			;25 cycles
	movlw 0x08
	movwf d1
Delay_0
	decfsz	d1, f
	goto	Delay_0

;*************************************
; Look up table
;;*************************************
DeteX
```

```
PageChange          Statement2
Statement2
            dt              "DeteX",0
StartMessageTwo
PageChange          Statement3
Statement3
            dt              "Press 1 to begin",0


;Barrel1
;PageChange         Statement4
;Statement4
;           dt              "Barrel#1 Type:T",0

;Barrel2
;PageChange         Statement5
;Statement5
;           dt              "Dist:1 Stat:F",0
;
;terminate
;PageChange         Statement6
;Statement6
;           dt              "Inspection Canceled"

Finish1
PageChange          Statement7
Statement7
            dt              "Inspection Done",0
Inspecting
PageChange          Statement8
Statement8
            dt  "Inspecting",0
Finish2
PageChange          Statement10
Statement10
            dt              "Press 1 for info",0
;Barrel3
;PageChange         Statement11
;Statement11
;           dt              "Dist:XXX Stat:",0
Barrel#
PageChange          Statement12
Statement12
            dt              "Barrel#",0
```

```
Types
PageChange          Statement13
Statement13
                dt              " Type:",0
Dist
PageChange          Statement14
Statement14
                dt              "Dist:",0
Stat
PageChange          Statement15
Statement15
                dt              " Stat:",0

Detected
PageChange      Statement16
Statement16
                dt              "Barrel Detected!",0
ReturningHome
PageChange          Statement17
Statement17
                dt              "Returning Home",0
;***************************************
; LCD control
;***************************************
Switch_Lines
                movlw B'11000000'
                call    WR_INS
                return

Clear_Display
                movlw B'00000001'
                call    WR_INS
                return

;***************************************
; Delay 0.5s
;***************************************
                delayL:
                movlw           0x30
                movwf           0x53
CONT3L:
                decfsz          0x53,f
                goto            CONT3L
                return
```

```
HalfS
        local     HalfS_0
    movlw 0x88
    movwf COUNTH
    movlw 0xBD
    movwf COUNTM
    movlw 0x03
    movwf COUNTL

HalfS_0
    decfsz COUNTH, f
    goto   $+2
    decfsz COUNTM, f
    goto   $+2
    decfsz COUNTL, f
    goto   HalfS_0

    goto $+1
    nop
    nop
                return

      END
```

15.4. Appendix D – AHP Decision Making Details

Table D1: Relative Preference with respect to Objective 1

	Design #1	Design #2	Design #3	Design #1	Design #2	Design #3	Overall Preference
	Relative Preference			Normalized Relative Preference			
Design #1	1	1	1	0.333	0.333	0.333	0.333
Design #2	1	1	1	0.333	0.333	0.333	0.333
Design #3	1	1	1	0.333	0.333	0.333	0.333

Since all the conceptual designs complete the task, they are preferred equally.

Table D2: Relative Preference with respect to Objective 2

	Design #1	Design #2	Design #3	Design #1	Design #2	Design #3	Overall Preference
	Relative Preference			Normalized Relative Preference			
Design #1	1	1/5	1/5	0.091	0.091	0.091	0.091
Design #2	5	1	1	0.455	0.455	0.455	0.455
Design #3	5	1	1	0.455	0.455	0.455	0.455

Due to the need of Design #1 being a particular size thereby avoiding the barrel during arm retraction, this design is harder to construct since it heavily relies on the dimensions to the success of the project. Due to this, the machine will be harder to construct and be more susceptible to measurement errors. Design #2 and #3 have similar construction difficulties with the only real difficulties arising from the need to have the level-detecting sensors are precise locations which is a need for all three designs.

Table D3: Relative Preference with respect to Objective 3

	Design #1	Design #2	Design #3	Design #1	Design #2	Design #3	Overall Preference
	Relative Preference			Normalized Relative Preference			
Design #1	1	1/5	1/5	0.091	0.091	0.091	0.091
Design #2	5	1	1	0.455	0.455	0.455	0.455
Design #3	5	1	1	0.455	0.455	0.455	0.455

Due to the need for Design #1 being a particular size, the design is less flexible to size changes. Designs 2 and 3 are not susceptible to the same limitation which is evidenced by the lack of dimensions listed in their conceptual design proposals.

Table D4: Relative Preference with respect to Objective 4

	Design #1	Design #2	Design #3	Design #1	Design #2	Design #3	Overall Preference
	Relative Preference			Normalized Relative Preference			
Design #1	1	2	2	0.500	0.500	0.500	0.500
Design #2	1/2	1	1	0.250	0.250	0.250	0.250
Design #3	1/2	1	1	0.250	0.250	0.250	0.250

Due to the inherent smaller size of Design #1 it has a slight edge in the cost of project department. It should be noted that the other Designs could also be made smaller if need be. Further cost analysis is done in the Subsystem Design Considerations and cost is attempted to be limited in the selection of wheels, sensors and actuators.

Table D5: Relative Preference with respect to Objective 5

	Design #1	Design #2	Design #3	Design #1	Design #2	Design #3	Overall Preference
	Relative Preference			Normalized Relative Preference			
Design #1	1	2	2	0.500	0.500	0.500	0.500
Design #2	1/2	1	1	0.250	0.250	0.250	0.250
Design #3	1/2	1	1	0.250	0.250	0.250	0.250

All design options clear the 10kg design constraint and this is ensured by the subsystem design considerations. As with RP w.r.t Objective 4, Design #1's need of a smaller size gives it a slight advantage in the weight department.

Table D6: Relative Preference with respect to Objective 6

	Design #1	Design #2	Design #3	Design #1	Design #2	Design #3	Overall Preference
	Relative Preference			Normalized Relative Preference			
Design #1	1	2	2	0.500	0.500	0.500	0.500
Design #2	1/2	1	1	0.250	0.250	0.250	0.250
Design #3	1/2	1	1	0.250	0.250	0.250	0.250

As with the weight, all designs ensure that the 55x55x55 cm volume constraint is met. Similarly, Design #1 has a slight advantage due to its smaller size.

Table D7: Relative Preference with respect to Objective 7

	Design #1	Design #2	Design #3	Design #1	Design #2	Design #3	Overall Preference
	Relative Preference			Normalized Relative Preference			
Design #1	1	1	1	0.333	0.333	0.333	0.333

Design #2	1	1	1	0.333	0.333	0.333	0.333
Design #3	1	1	1	0.333	0.333	0.333	0.333

The time dependence of each design is determined by the sensor time and speed of travel which is determined in the subsystem design and is independent of the system design.

Table D8: Relative Preference with respect to Objective 8

	Design #1	Design #2	Design #3	Design #1	Design #2	Design #3	Overall Preference
	Relative Preference			Normalized Relative Preference			
Design #1	1	1/4	1/3	0.125	0.130	0.118	0.124
Design #2	4	1	3/2	0.500	0.522	0.529	0.517
Design #3	3	2/3	1	0.375	0.348	0.353	0.359

Due to the asymmetrical nature of the need for the Proximity sensor for column detection, Design #1 suffered in the Aesthetic Appeal department. Design #2 slightly out performed Design #3 due to its simplicity, symmetry in having sensors at both the front and back of the machine and freedom of shelf implementation and openness for LCD and keypad integration.

Table D9: Relative Preference with respect to Objective 9

	Design #1	Design #2	Design #3	Design #1	Design #2	Design #3	Overall Preference
	Relative Preference			Normalized Relative Preference			
Design #1	1	8	10	0.816	0.857	0.714	0.796
Design #2	1/8	1	3	0.102	0.107	0.214	0.141
Design #3	1/10	1/3	1	0.082	0.036	0.071	0.063

Design #1 is preferred in terms of Extra Design Features due to its smaller size which could potentially cover the Compactness and Portability extra design feature. All designs could implement Real-time Date/Time Display, Permanent Logs, PC Interface and Remote Operation due to the heavy reliance of the microcontroller for such tasks, with Remote Operation requiring a sensor which is small enough to implement in all designs.

After computing the overall preference of each design with respect to each objective, the relative importance of each objective with respect to the other objectives was determined as displayed in TableA10 below.

Table D10: Relative Importance Mapping

	Obj. 1	Obj. 2	Obj. 3	Obj. 4	Obj. 5	Obj. 6	Obj. 7	Obj. 8	Obj.9
Obj. 1	1	2	2	4	6	6	6	7	8
Obj. 2	1/2	1	1	2	3	3	3	3.5	4
Obj. 3	1/2	1	1	2	3	3	3	3.5	4
Obj. 4	1/4	1/2	1/2	1	3/2	3/2	3/2	7/4	2
Obj. 5	1/6	1/3	1/3	2/3	1	1	1	7/6	4/3
Obj. 6	1/6	1/3	1/3	2/3	1	1	1	7/6	4/3
Obj.7	1/6	1/3	1/2	2/3	1	1	1	7/6	4/3
Obj.8	1/7	2/7	2/7	4/7	6/7	6/7	6/7	1	9/8
Obj.9	1/8	1/4	1/4	1/2	3/4	3/4	3/4	8/9	1

These results were then normalized along each column to determine the relative overall importance as indicated in the rightmost column.

Table D11: Normalized Relative Importance Matrix

	Obj. 1	Obj. 2	Obj. 3	Obj. 4	Obj. 5	Obj. 6	Obj. 7	Obj. 8	Obj.9	Overall Importance
Obj. 1	0.331	0.331	0.322	0.331	0.331	0.331	0.331	0.331	0.332	0.330
Obj. 2	0.166	0.166	0.161	0.166	0.166	0.099	0.166	0.166	0.166	0.158
Obj. 3	0.166	0.166	0.161	0.166	0.166	0.110	0.166	0.166	0.166	0.159
Obj. 4	0.083	0.083	0.081	0.083	0.083	0.062	0.083	0.083	0.083	0.080
Obj. 5	0.055	0.055	0.054	0.055	0.055	0.043	0.055	0.055	0.055	0.054
Obj. 6	0.055	0.055	0.054	0.055	0.055	0.045	0.055	0.055	0.055	0.054
Obj.7	0.055	0.055	0.081	0.055	0.055	0.047	0.055	0.055	0.055	0.057
Obj.8	0.047	0.047	0.046	0.047	0.047	0.042	0.047	0.047	0.047	0.047
Obj.9	0.041	0.041	0.040	0.041	0.041	0.038	0.041	0.042	0.041	0.041

After normalization of the columns, the RP/RI Matrices for all objectives were computed as displayed in Tables A12 to A20 below. Furthermore the Consistency Index of each matrix was computed to ensure that it was under the 0.10 threshold value. It is evident from the Constancy Index being 0.00 for most of our matrices that our decision making system was consistent and that the choice of Design #2 for our system design is justifiable since it had the best overall performance throughout most of the objectives.

Table D12: Matrix with Respect to Objective #1

	Design #1	Design #2	Design #3	Eigenvalues

Design #1	1.00	1.00	1.00	0
Design #2	1.00	1.00	1.00	0
Design #3	1.00	1.00	1.00	**3**

$$Consistancy\ Index\ (CI) = 0.00$$

Table A13: Matrix with Respect to Objective #2

	Design #1	Design #2	Design #3	Eigenvalues
Design #1	1.00	0.20	0.20	**3**
Design #2	5.00	1.00	1.00	0
Design #3	5.00	1.00	1.00	0

$$Consistancy\ Index\ (CI) = 0.00$$

Table A14: Matrix with Respect to Objective #3

	Design #1	Design #2	Design #3	Eigenvalues
Design #1	1.00	0.20	0.20	**3**
Design #2	5.00	1.00	1.00	0
Design #3	5.00	1.00	1.00	0

$$Consistancy\ Index\ (CI) = 0.00$$

Table A15: Matrix with Respect to Objective #4

	Design #1	Design #2	Design #3	Eigenvalues
Design #1	1.00	2.00	2.00	**3**
Design #2	0.50	1.00	1.00	0
Design #3	0.50	1.00	1.00	0

$$Consistancy\ Index\ (CI) = 0.00$$

Table A16: Matrix with Respect to Objective #5

	Design #1	Design #2	Design #3	Eigenvalues
Design #1	1.00	2.00	2.00	**3**
Design #2	0.50	1.00	1.00	0
Design #3	0.50	1.00	1.00	0

$$Consistancy\ Index\ (CI) = 0.00$$

Table A17: Matrix with Respect to Objective #6

	Design #1	Design #2	Design #3	Eigenvalues
Design #1	1.00	2.00	2.00	**3**
Design #2	0.50	1.00	1.00	0
Design #3	0.50	1.00	1.00	0

$$Consistancy\ Index\ (CI) = 0.00$$

Table A18: Matrix with Respect to Objective #7

	Design #1	Design #2	Design #3	Eigenvalues
Design #1	1.00	1.00	1.00	0
Design #2	1.00	1.00	1.00	0
Design #3	1.00	1.00	1.00	3

$$Consistancy\ Index\ (CI) = 0.00$$

Table A19: Matrix with Respect to Objective #8

	Design #1	Design #2	Design #3	Eigenvalues
Design #1	1.00	0.24	0.35	**2.82**
Design #2	4.16	1.00	0.69	0.00
Design #3	2.88	0.69	1.00	0.19

$$Consistancy\ Index\ (CI) = -0.09$$

Table A20: Matrix with Respect to Objective #9

	Design #1	Design #2	Design #3	Eigenvalues
Design #1	1.00	5.64	12.65	**3.0117**
Design #2	0.18	1.00	2.24	-0.01 + 0.003i
Design #3	0.08	0.45	1.00	-0.01 – 0.003i

$$Consistancy\ Index\ (CI) = 0.01$$