ECE 4950 Integrated System Design 1

CUSTOMER REQUIREMENTS AND FINAL DESIGN PARAMETERS

December 12th , 2022

Team 16

Matthew Wehrly, Deeb Armaly, Brandon White, Darius Givens, Tre’zjon Bell

**Executive Summary**

The objective of the final lab was to consolidate the design of the autonomous robot to complete the game board competition. The robot had to be financially budgetable for the customer and satisfied all their requirements. The main components of the user interface operated robot included an Arduino, a vision-enabled camera, a closed-loop PID controller, an encoder, and an emergency stop button. The GUI was designed to have a simplistic and easily understandable navigation system while also being aesthetically pleasing. No batteries were used in the design as instructed and degrees of freedom are allowed with rotational movement. The robot was designed to be reliable, durable, safe, and produces low noise. Due to competition, the robot was built with the intent to be competitively fast, efficient, and game board. comprehensible.

**Engineering Requirements**

| Customer Requirements | Engineering Requirements | Testing/Verification |
|---|---|---|
| Color Detection | Use of very distinguishable colors in a well-lit environment with little to no coloration | Used a series of tests to determine the accuracy of color detection |
| Position Detection | Adjusting and testing camera positioning | Used a series of tests to determine the accuracy of position detection |
| Game Piece Detection | Proper programming design, and well-lit environment with little to no coloration | Detect up to nine positions of colored stickers, simultaneously |
| Clock Hand Mobility | Clock Hand must have degrees of freedom from the placement on the shaft so that the hand can point to any position on the game when the shaft spins. | Rotate the clock hand 360 degrees and use the two-dimensional coordinate plane to test which axis can move freely. |
| Motor Loop Controlled Subsystem | The motor should have code of its own to allow private functionality using real time response to the motion starting from the team's choice. | Matlab can test separate parts of the game by sectioning off the code to run once at a time to verify each subsystem works correctly independently. |
| Shaft Mechanization | Upon selection, the mounted arm will turn (clockwise/counterclockwise) and in one smooth motion come to rest at the centroid of the required color. | Code from Matlab can determine whether to move positive sequence or negative sequence. Also noting how fast the shaft will move until the shaft stops at the correct color from the user interface. |
| Efficient | The sticker colors and positions on the game board will have been processed using the game state sensor and Matlab image processing. | The sensor should have color detection from the camera to make sure the right color was processed through the image using guided parameters from the team. |
| User-Friendly | A sticker color will be chosen by the user in Matlab either from a command prompt, a menu or a GUI consisting of at least simple push buttons or something similar. | The current code will allow the user to pick from the list of colors given and the collected responses should keep a tab in order of those picked colors. |

| | | |
|---|---|---|
| Productivity | The quality of the user interface and mounted arm motion (directly correlated to motor control) will be used to judge the implementation. | A clock timer can time how fast the mover responds to the code as the user inputs the parameters to compare to real time. |
| Color Proximity Detection | In the case of multiple stickers of the same color, the mounted arm must come to rest at the closest chosen colored sticker. | A conditional statement will be written to ensure that in the instance multiple colors come up that the first available color will be picked first based on its proximity to the shaft. |
| Cost Effective ($300) | Built in totality while satisfying requirements with given budget | Used lab equipment and kept track of expenses to accommodate spending limit |
| Reliable | Handled game board pieces accordingly to game rules by following location assignments | Used a series of tests to determine the accuracy of washer location identification and desired displacement |
| Durable | Designed to manage moving multiple washer pieces for extensive game play | Used a series of tests to determine the durability of robot actuator during play time |
| Safe | Controlled movement to avoid dangerous workplace conditions | Wires and electrical components were enclosed to remove potential electrical/mechanical hazards |
| Low Noise Production | Controlled noise output to reduce noise pollution | Ran under stressful computational conditions to achieve worst case scenario sound production |
| Autonomous | Designed to operate with human interaction | Programmed and wired appropriately to enable desired rotational movement and game piece placement/retrieval |
| Degrees of Freedom Enabled Movement | The robot should move with at least one degree of freedom | Designed with limited free movement capability |

**Design Details**

The construction and configuration of the robotic system can be explained with attention drawn to 4 different main design integrations: electrical wiring, closed-loop PID control, camera vision, and user interface. The wiring was purposeful and designed to enable rotational movement of the robotic arm, power the circuit components including the Arduino, stop button, encoder, and motor driver while establishing signal communication between the devices. The closed-loop control system is driven by the motor model simulation. The camera vision and user interface is computed using Matlab programming and software.

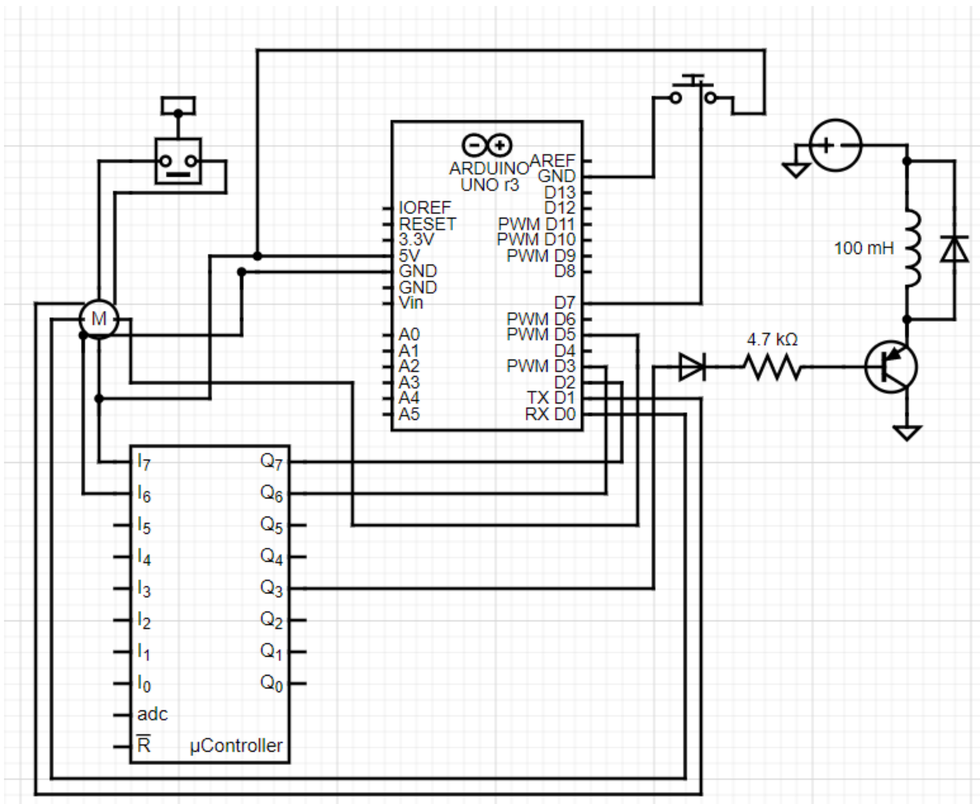


Figure 1: Circuit Diagram

A circuit schematic of the washer robot machine is provided in Figure 1 above. Due to limited resources, the schematic doesn't exactly illustrate the pinouts of the actual devices used, although similarities are apparent. In a clockwise rotational explanation, starting at the top left corner, the robot head-like device shown represents the power source box that is used as the primary power source of the machine. The Arduino shown is an Uno model while the Arduino used in the lab is a Due model. However, the difference in Arduino models is noted for accurate documentation and has no effect on the understanding of the device's overall functionality. In regards to the digital pins of the Uno model, TX D0 and RX D1 represent D30 and D31 and D7 represents communication pin D21 on the Due model respectively. To the right of the Arduino is the pushbutton and the circuitry of the electromagnet that is connected to the encoder. Due to the omission of an encoder component in the schematic drawing software, a microcontroller was used to represent the device. In regards to the pins of the microcontroller I6 and I7 represent power and ground and Q6 and Q7 represent Hall sensors A and B respectively of the encoder device. Lastly, the motor is represented by the “M” labeled component shown and lacks an adequate pinout. Consequently, the pinout connections will be stated explicitly for accurate documentation.

The circuit wiring is the lowest level of the system architecture. In clockwise rotation, starting with the power source box, 12V power and ground is supplied to the motor driver. The motor drive is enabled through pin 5, a digital PWM input pin of the Arduino. A PWM pin is used to control the duty cycle of the motor driver. Digital pins 30 and 31 of the Arduino drive the logic of the motor and are connected to pins INT3 and INT4. The motor driver supplies power to the encoder that enables the movement of the robot arm. In addition, the Arduino is also connected to the encoder using digital pins 2 and 3 as interrupt handlers. The electromagnet is connected to the encoder and used as the actuator for moving game pieces. Lastly, the Arduino is connected to the conditional switch of the stop push button using communication pin 21 to incorporate a safety measure into the design.

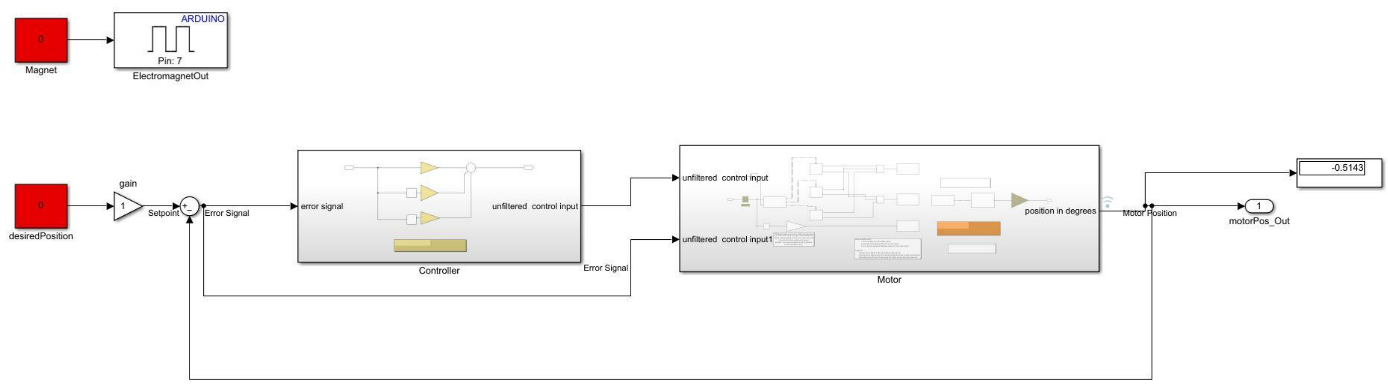


Figure 2: Motor Simulation

The next level of the system architecture is the motor simulation in Simulink shown in Figure 2 above. Using a bottom up approach, the motor simulation can be explained with attention to detail of the controller and the motor makeup. The motor is divided into several inner elements including the motor direction and counter along with other small intricate pieces that are together used to calculate angular position in degrees. The PID controller by nature controls the motor through a modification of the proportional, derivative ,and integral gain values. These gains work to correct the error of movement heading in the wrong direction and reduce the instance overtime, act as a catalyst to increase the time of reaction in the instance of error, and denote the instance of an error increasing over time respectively. These values were toggled with for the right values to rotate the robotic arm with smooth control. The computation of these values enables the output of the arm position and thus is how the robot is engineered to move game pieces between board locations.

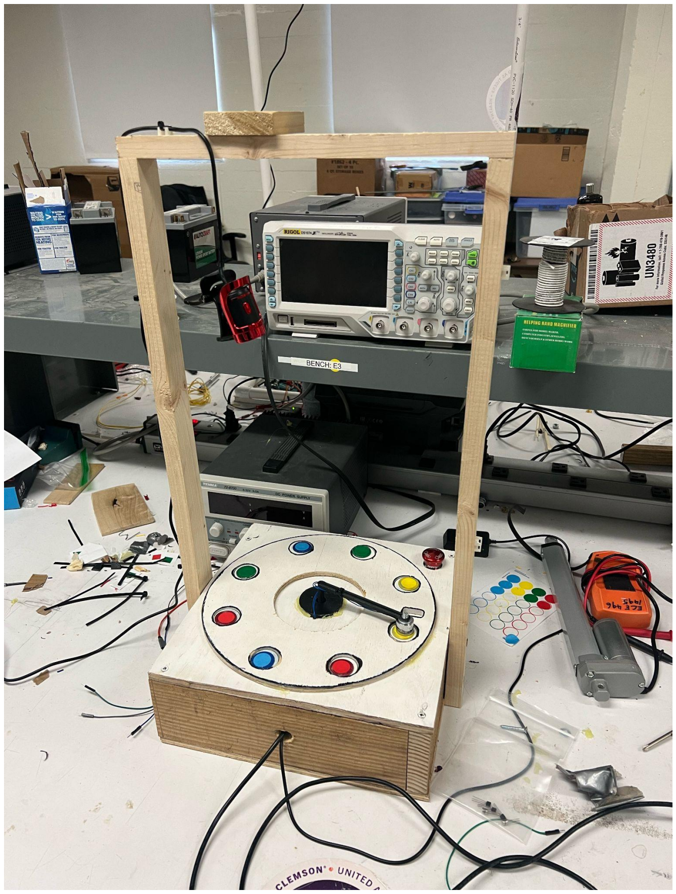

Figure 3: Final Design

The third level of system architecture is the final design of the robot that combines the inner circuit wiring and the software computation tools to enable autonomous capabilities as shown above in Figure 3. The design of the arm was constructed with intent to move the arm, using the position calculations from the encoder, to find and move game pieces accordingly around the board in a rotational manner. Starting at the home location, the robot first uses a vision enabled camera to gather information about the position of game pieces on the board. The image is then processed and game pieces are moved one at a time with a single motor. Due to the varied strength of the electromagnetic, small whole pucks were 3D printed for accommodation that allows pieces to be picked up with closer proximity to the magnetic pull. After every game play, the arm returns back to the home position if not interrupted by the emergency stop button.

While this design completed the tasks of game play, there were limitations that were accounted for that would have made a more suitable design for efficiency and timing. The use of a singular motor and a rotational robot arm was limiting in the sense that it didn’t allow for the use of a rotational base and the use of two motors for more efficient task completion. The use of a rotating board base would avoid the risk of washers falling due to the speed and inertia of the arm movement. In addition, the use of a second servo motor would simplify the retrieval and drop off of game pieces, limiting the arm to only an up and down position movement that would increase the safety and timing of each game piece placement.

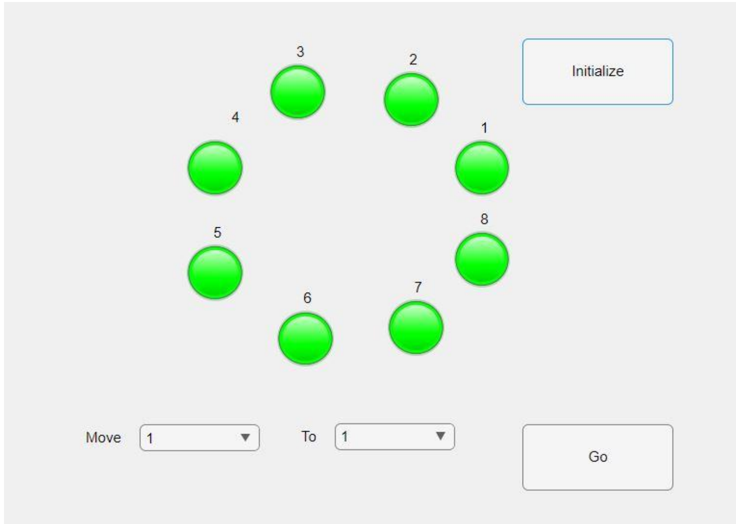

Figure 4: GUI

The fourth and topmost level of system architecture is the GUI, graphical user interface as shown above in Figure 4. The user interface is enabled through the vision sensors embedded in the camera that overlooks the game board. At start time, the game pieces will turn blue, yellow, green, or red to denote the color of the game piece present at each location and a color of black to denote the absence of a game piece in a particular location. Consequently, when moving pieces, the user can use the two drop down menus to change the initial location to the desired location to place the game piece. The resulting board will then be updated and the move game piece and black spot visually swap places in the interface. This allows the user to keep track of the game board without looking directly at the pieces and makes an adaptable, user friendly game play experience.

**Analysis of Final Prototype Performance**

| **Customer Requirements** | **Engineering Requirements** | **Testing/Verification** | **Performance Analysis** |
|---|---|---|---|
| Color Detection | Use of very distinguishable colors in a well-lit environment with little to no coloration | Used a series of tests to determine the accuracy of color detection | Detects colors accurately |
| Position Detection | Adjusting and testing camera positioning | Used a series of tests to determine the accuracy of position detection | Detects positions accurately |
| Game Piece Detection | Proper programming design, and well-lit environment with little | Detect up to nine positions of colored stickers, simultaneously | Detect absence and presence of game pieces accurately |

| | | | |
|---|---|---|---|
| | to no coloration | | |
| Clock Hand Mobility | Clock Hand must have degrees of freedom from the placement on the shaft so that the hand can point to any position on the game when the shaft spins. | Rotate the clock hand 360 degrees and use the two-dimensional coordinate plane to test which axis can move freely. | Smooth and accurate, but slow movement |
| Motor Loop Controlled Subsystem | The motor should have code of its own to allow private functionality using real time response to the motion starting from the team's choice. | Matlab can test separate parts of the game by sectioning off the code to run once at a time to verify each subsystem works correctly independently. | Controlled and accurate calculation of game piece placement |
| Shaft Mechanization | Upon selection, the mounted arm will turn (clockwise/counterclockwise) and in one smooth motion come to rest at the centroid of the required color. | Code from Matlab can determine whether to move positive sequence or negative sequence. Also noting how fast the shaft will move until the shaft stops at the correct color from the user interface. | Allowance of mechanical rotation of robotic arm |
| Efficient | The sticker colors and positions on the game board will have been processed using the game state sensor and Matlab image processing. | The sensor should have color detection from the camera to make sure the right color was processed through the image using guided parameters from the team. | Lacks fast movement but assures correct game piece placement |
| User-Friendly | A sticker color will be chosen by the user in Matlab either from a command prompt, a menu or a GUI consisting of at least simple push buttons or something similar. | The current code will allow the user to pick from the list of colors given and the collected responses should keep a tab in order of those picked colors. | Aesthetically pleasing and easily understood digital representation of game play |
| Productivity | The quality of the user interface and mounted arm motion (directly correlated to motor | A clock timer can time how fast the mover responds to the code as the user inputs the | Correct adherence to rules of game play |

| | | | |
|---|---|---|---|
| | control) will be used to judge the implementation. | parameters to compare to real time. | |
| Color Proximity Detection | In the case of multiple stickers of the same color, the mounted arm must come to rest at the closest chosen colored sticker. | A conditional statement will be written to ensure that in the instance multiple colors come up that the first available color will be picked first based on its proximity to the shaft. | Accurate detection of desired game piece placement |
| Cost Effective ($300) | Built in totality while satisfying requirements with given budget | Used lab equipment and kept track of expenses to accommodate spending limit | Meets financial budget expectations |
| Reliability | Handled game board pieces accordingly to game rules by following location assignments | Used a series of tests to determine the accuracy of washer location identification and desired displacement | Accurate at playing game board according to rules and regulations |
| Durability | Designed to manage moving multiple washer pieces for extensive game play | Used a series of tests to determine the durability of robot actuator during play time | Handles movement of multiple game pieces for extended game play time |
| Safety | Controlled movement to avoid dangerous workplace conditions | Wires and electrical components were enclosed to remove potential electrical/mechanical hazards | Safe, slow movement of arm that lacks hazardous rotational movements |
| Noise Cancellation | Controlled noise output to reduce noise pollution | Ran under stressful computational conditions to achieve worst case scenario sound production | Low noise producing machinery for suitable workplace conditions |
| Autonomous Capability | Designed to operate with human interaction | Programmed and wired appropriately to enable desired rotational movement and game piece placement/retrieval | Plays game board without human interference aside form graphical interface |
| Degrees of Freedom | The robot should move | Designed with limited | Movement is enabled |

| | | | |
|---|---|---|---|
| Enabled Movemented | with at least one degree of freedom | free movement capability | with required degree of freedom |

**Project Schedule/Gantt Chart**

The schedule for project 1 is displayed in Figure 5 below:

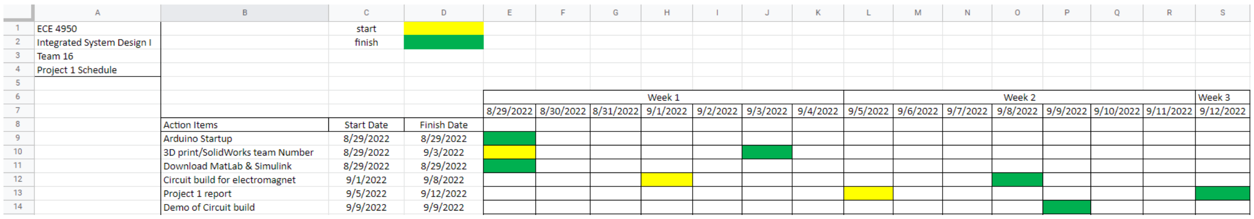


Figure 5: Project 1 Schedule

The schedule for project 2 is displayed in Figure 6 below:

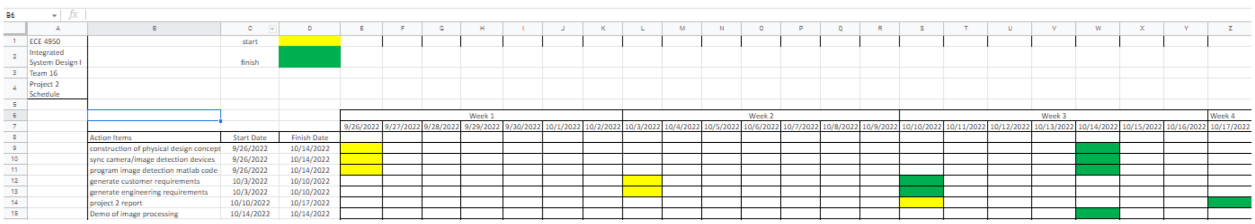


Figure 6: Project 2 Schedule

The schedule for project 3 is displayed in Figure 7 below:

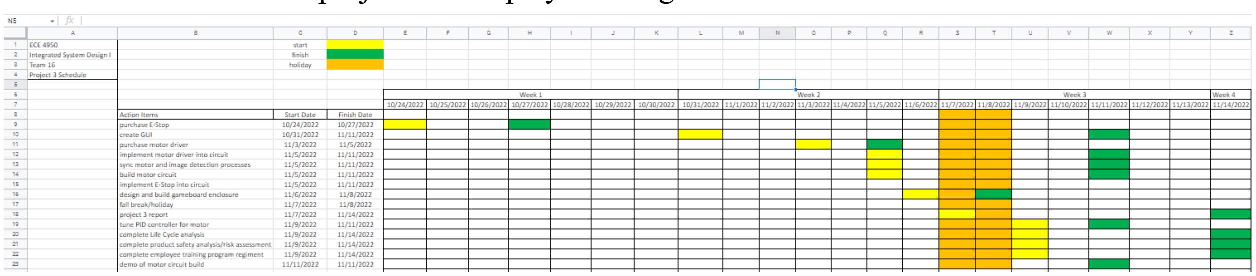


Figure 7: Project 3 Schedule

The schedule for project 4 is displayed in Figure 8 below:

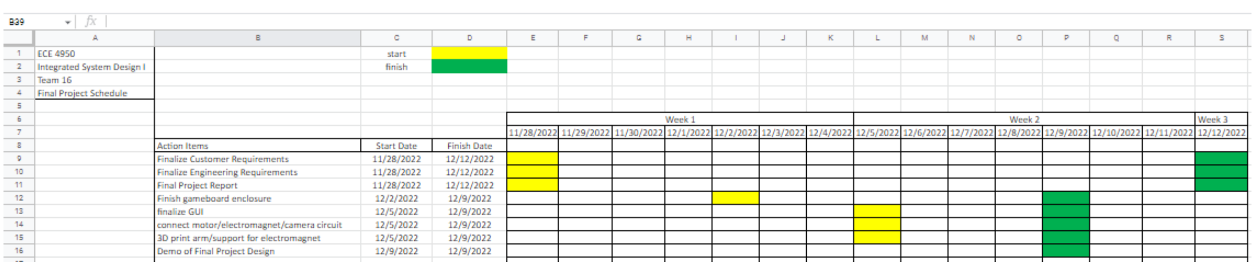


Figure 8: Project 4 Schedule

**Website**

- https://sites.google.com/g.clemson.edu/thewashermovers/home?authuser=1

**References**


- Understanding PWM (**Understanding PWM – EBLDC.COM**)


## ECE 4950 Project 4 – Customer Requirements and Final Design Parameters

Use the guidelines below to complete your report and add at the end of your report.
Group Member Last Names:______________________________________________________

| Score | Pts | | Performance Indicators |
|---|---|---|---|
| | 5 | **General Format - Professional Looking Document/Preparation (whole document)**<br>a) Fonts, margins (11pt, times new roman, single spaced. 1" margins on all sides).<br>b) Spelling and grammar are correct<br>c) Layout of pictures – all figures need numbers and captions and must be referenced in the text<br>d) Follows the page limitations below.<br>e) References. Use IEEE reference format.<br>f) This grading sheet is included as the final page. | g.1 |
| | 0 | **Page 1: Title, Group Name, Group Members, and Date**<br>**Executive Summary** (1 concise, well-written paragraph)<br>Provide an overview of this project. Briefly describe what you did and what you learned. | g.1 |
| | 5 | **Page 2: Engineering Requirements** (<1 page)<br>Bulleted list of Final Design Engineering Requirements | e.1 |
| | 10 | **Pages: 3-7: Design Details (<5 pages)**<br>Describe a system that can be built including System Architecture and System Integration based on the Engineering Requirements. Do not include data sheets or software code. | e.2 |
| | 10 | **Page 8: Analysis of Final Prototype Performance** (<1 page)<br>Did it succeed or fail to meet customer requirements? What went wrong and what happened in the design process to allow this problem? Make a table of the customer requirements and address how well your design met these expectations. | e.3<br>e.4<br>i.1 |
| | 5 | **Page 9: Project Schedule/Gantt Chart** (<1 page)<br>Create a schedule (Gantt chart) that shows the tasks and schedule for your project. Start from the very beginning of your project and extend to the end (completing final report and presentation). | k.2 |
| | | **Page 10** This grading sheet is included as the final page. | |
| | 50 | Laboratory demonstration of your prototype (evaluated by instructor and TAs). Evaluator will manipulate the interface and evaluate how well the system provides the timing and display functions (i.e. how well does the closed loop control work). Is it well built? Neat wiring? (.6 * the prototype evaluation score) | g.2 |
| | 15 | Rating by reviewers during competition | g.2 |