ECE Project 6, Final Project:
Whack-a-Mole

Group 3 "No Induction Needed"
Chandler Coffman, Joshua Johnston, Chase Lanier, Justin McAllister, Anthony Stone
http://people.clemson.edu/~jlj8/index.html
9 December 2015

Objective
Design a fast moving, quick problem solving, autonomous robot to compete against other team robots.

Customer Requirements
Overall System requirements should always agree to rules defined within the game and its cost shall not exceed $300.

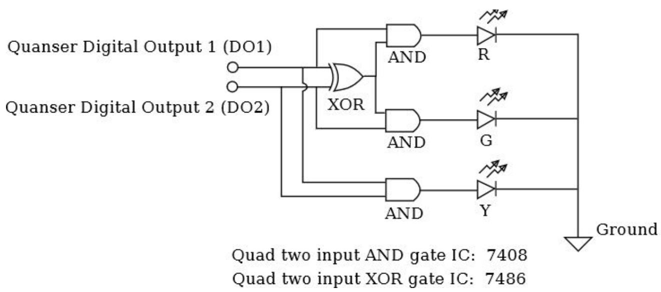
The physical system must be kept to the computing interfaces, electrical and mechanical design pieces and constructed elements specific to the unit. The xPC workstation, Q4 boards and Techron amplifying systems are to be considered part of this unit. The system's overall power must come from a traditional AC power terminal.

The systems performance must be reliable, durable and safe. Noise levels must be minimal and there should be no harmful parts or actions. The system must maintain no less than one degree of freedom and the xPC Target maintains the closed-loop position control. Electric and electronic circuits should be built from simple and basic components. The unit should run autonomously and act quickly in problem solving. The system should be user friendly and provide a Graphical User Interface during gameplay operation.

System Integration

Electrical Circuit Design

The LED circuit logic uses 3 AND gates and 1 XOR gate per section of LEDs. The inputs for the logic come from the 16 digital output pins on the Quanser board. The logic can be shown along with truth table below.



Truth Table

DO1	DO2	R	G	Y
0	0	0	0	0
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

Figure 1a & 1b: Schematic of Digital Output gates(left) and the truth table logic (right)

The lights used on the game board are low power LEDs requiring about 1.8V and 2mA. We used the provided power supply giving 5V to the ICs in this circuit, the LEDs, and the push button circuit. This meant that we were putting 5V over the LEDs, which was pretty high for the LEDs. To reduce the voltage to 2.5V we added resistors. We added 1500Ω resistors after the LEDs, which reduced the voltage across the LEDs too much. Instead we increased the power supply to 5.5V to make them a little brighter. The wiring diagram for the LED circuit is shown below.

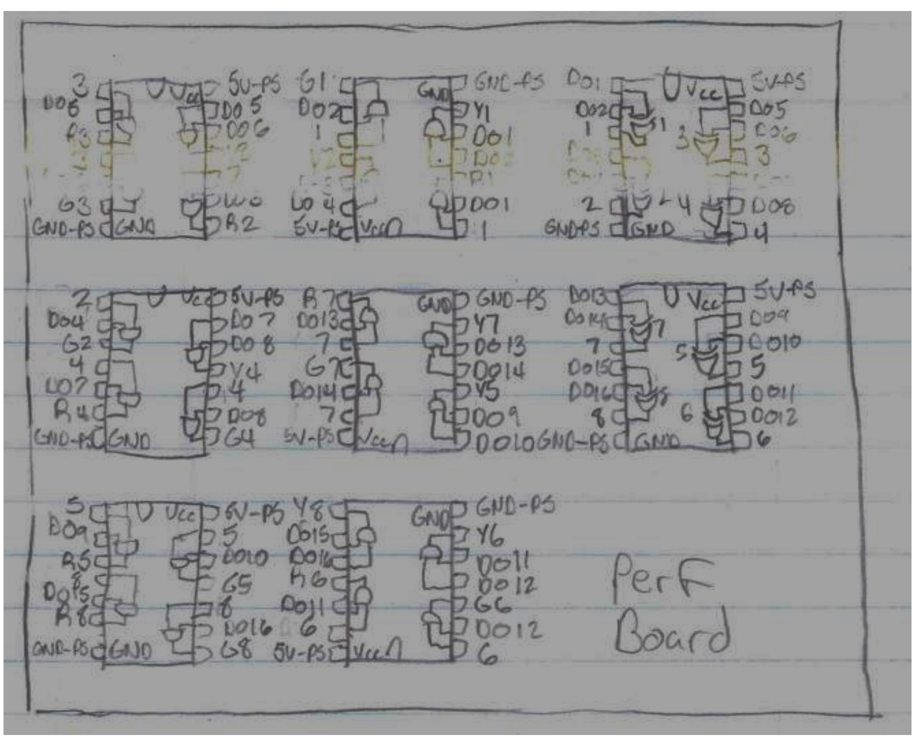


Figure 2: LED wiring schematic

The supply voltage of 5.5V, from the power supply, was used for the push button. Each push button has the same value resistor between each button. The resistor used was 470Ω. This wiring design enabled detection of which button is pressed with ease. The circuit is a voltage divider, providing a different voltage to the analog input for each button. The push button circuit and voltage table is shown below.

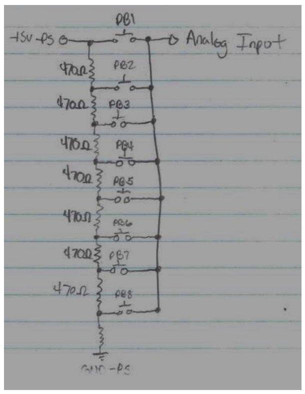


Figure 3: Pushbutton wiring schematic

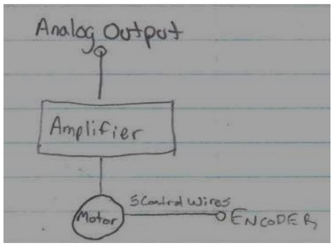
Action	V_{out}
PB1	1.000 Vs
PB2	0.875 Vs
PB3	0.750 Vs
PB4	0.625 Vs
PB5	0.500 Vs
PB6	0.375 Vs
PB7	0.250 Vs
PB8	0.125 Vs

Figure4: Table of supplied voltage values

The Whack-a-Mole logic block was used with a provided Simulink system. The basic logic behind this block maintains 5 inputs and 4 outputs are provided with descriptions.

- Switch_Vin reads the push buttons as an analog input channel.
- Switch_Vs is the push button's supplied voltage and remains constant.
- Switch_Dir determines the order of the push buttons and allows the each pusbutton's position in the sequence to be reversed such that 8 can be 1, 7 can be 2, etc.
- QRNum is one digit that demonstrates the number of the round and version of sequence during the final demonstration. It will provide the LEDs lighting sequence and time.
- QRSeq defines the color sequence for certain rounds
- Red_8BitOutput is connected to channels 1 through 8 of the Digital Output (DO) block and is a signal to notify a Red LED ON.
- Green_8BitOutput is connected to channels 9 through 16 of the Digital Output (DO) block and is a signal to notify a Green LED ON.
- True Hit Count accounts for the correct color's push button "hit" within the appropriate time limit.
- Stop Flag is the output flag showing the sequence's end.

To power our motor and solenoid we used two analog outputs that ran to our amplifier. These outputted to our loads because the Quanser board cannot supply the needed current. The motor has a 5 pin control that is plugged into the encoder on the Quanser board. The circuit for the motor and solenoid are below.



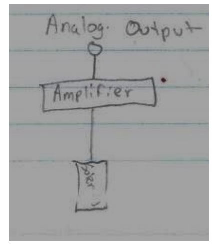


Figure 5a & 5b: Diagrams illustrates motor and solenoid output and amplifier connections

The final circuit was for safety. If trouble arose, or anything went wrong, the emergency stop button would be handy. 5V from the power supply was wired into the button. We wired the other terminal of the button to the analog input so when the button was pressed 5V could be read into the analog input. The circuit for this is shown below.

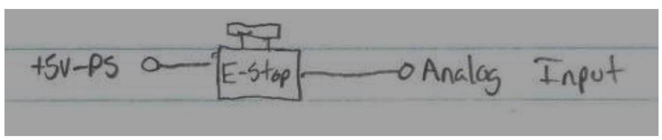


Figure 6: E-stop wiring schematic

Programming and Software Design

Error Correction Function

When running the robot, the group discovered that after moving to several positions the motor error began to accumulate. This caused the arm's accuracy when moving to the buttons to degrade until eventually buttons were missed. In order to accommodate for this we implemented a matlab function that would correct the motor based on the error signal it received. This function works by moving the motor to the angle that is passed into the function. The function then measures the error signal that it output by the motor, and makes the motor move to the original angle passed into the function plus the measured error signal. Moving the motor with this function instead of just setting the parameter in Simulink caused the arm to maintain accuracy for a longer period of time.
General Code Description

All of the code for the robot was triggered through the GUI code that was given to us by the instructors. We modified the code to include the buttons and information required for the project, and to call the functions necessary for robot operation. The GUI was capable of calling two functions. One of the functions was used to scan the QR code for the round and sequence information. The other function, whackaMole, was used to operate the robot based on the scanned QR code. whackaMole worked by first calculating the number of lights from the QR information passed into it. Next it loaded a background image of the game board into the matlab workspace. This image would be used over the course of the game for image processing. The function will then start the Simulink model that is loaded onto the Q4 board before going into a loop that controlled the robot during the game. Within the loop there are separate conditions for the different possible rounds of the game. No matter which round of the game was being played, the next step of the function was to go into an image processing loop. This loop would continue to take pictures and perform background subtraction with some area filtering, until a light was detected. Once the function thinks it has located a light, it passes the X and Y pixels into another function to verify it is a valid light location. If the location is not valid for a light the function will return to taking pictures until it locates another possible light, otherwise the function will decide if it needs to move the arm of the robot or not. The conditions for movement were different depending on the round number. If the proper conditions were met, the X and Y coordinates of the light were passed into another function to determine which sector of the board the light was in and its color. Another auxiliary function was used to determine the proper motor angle based on the sector of the light. The color parameter was only used in rounds two through four. After the motor angle has been determined the function will move the motor to the specified angle using the error correcting movement function. The final step in the function is to fire the solenoid to turn of the illuminated light. This process will continue until all lights in the QR sequence have been hit.

Code Changes for Different Rounds

In order to account for the different rounds of the game, the program had conditional statements that would execute different code based on the round number in the QR code. Since there were two different sets of rules, one for round one and another for rounds two through four, this could be accomplished with two different conditional statements. The first conditional statement, which was used for round one, would move the arm of the robot and fire the solenoid if the robot perceived any light on the board. However, the second conditional statement would only move the motor and fire the solenoid if the light was in a valid location and if the color parameter matched the desire color in the QR code.

Models and Circuits for Simulink

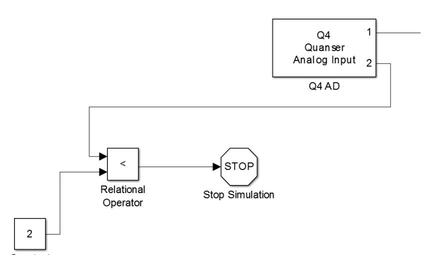


Figure 7: Simulink E-Stop Integration model.

```
function[] = moveMotor(motorAngle, angle, tg)
        tg.setparam(motorAngle, angle);
        error = tg.getsignal(3);
        angle = tg.getsignal(4) + error;
        tg.setparam(motorAngle, angle);
end
```

Figure 8: Function explaining how the motor moves specific distances.

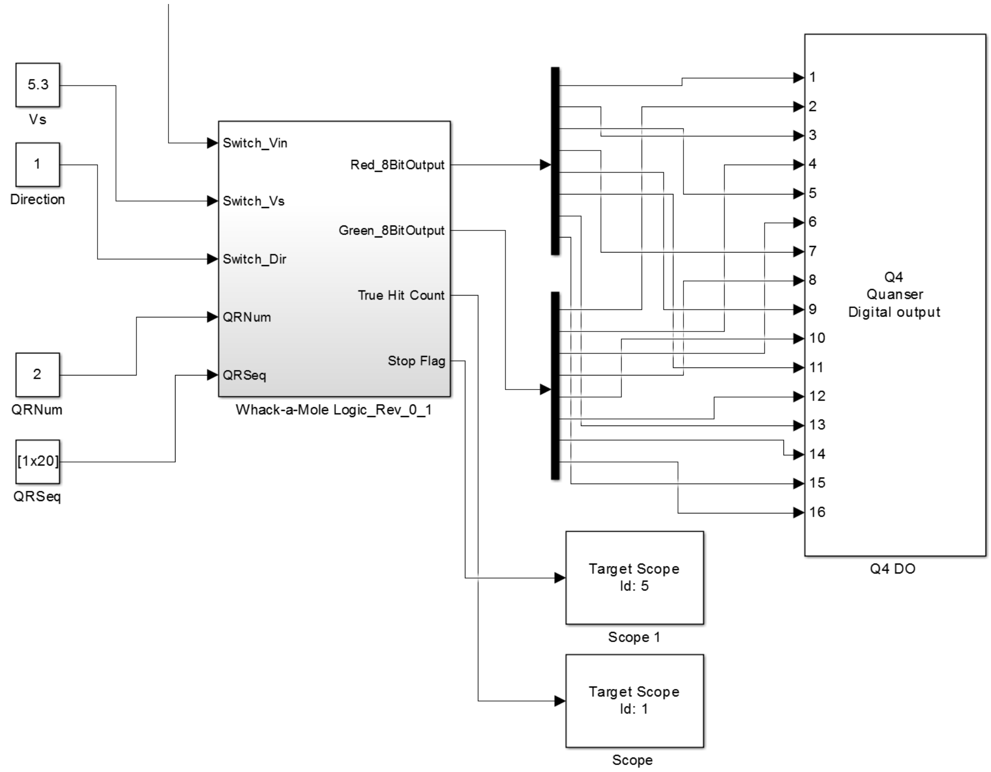


Figure 9:Simulink lighting circuit and model.

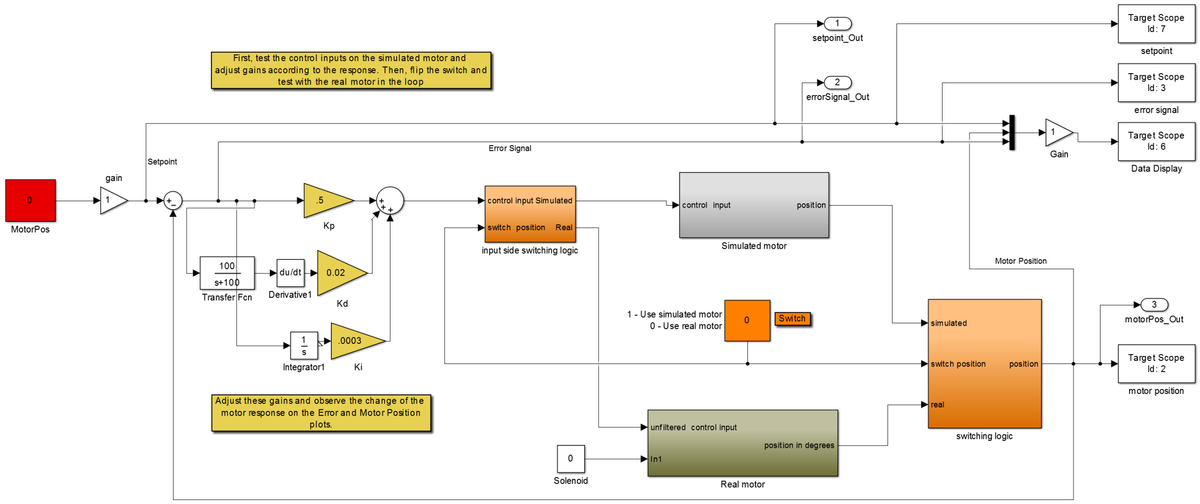


Figure 10: Simulink Motor Circuit model.

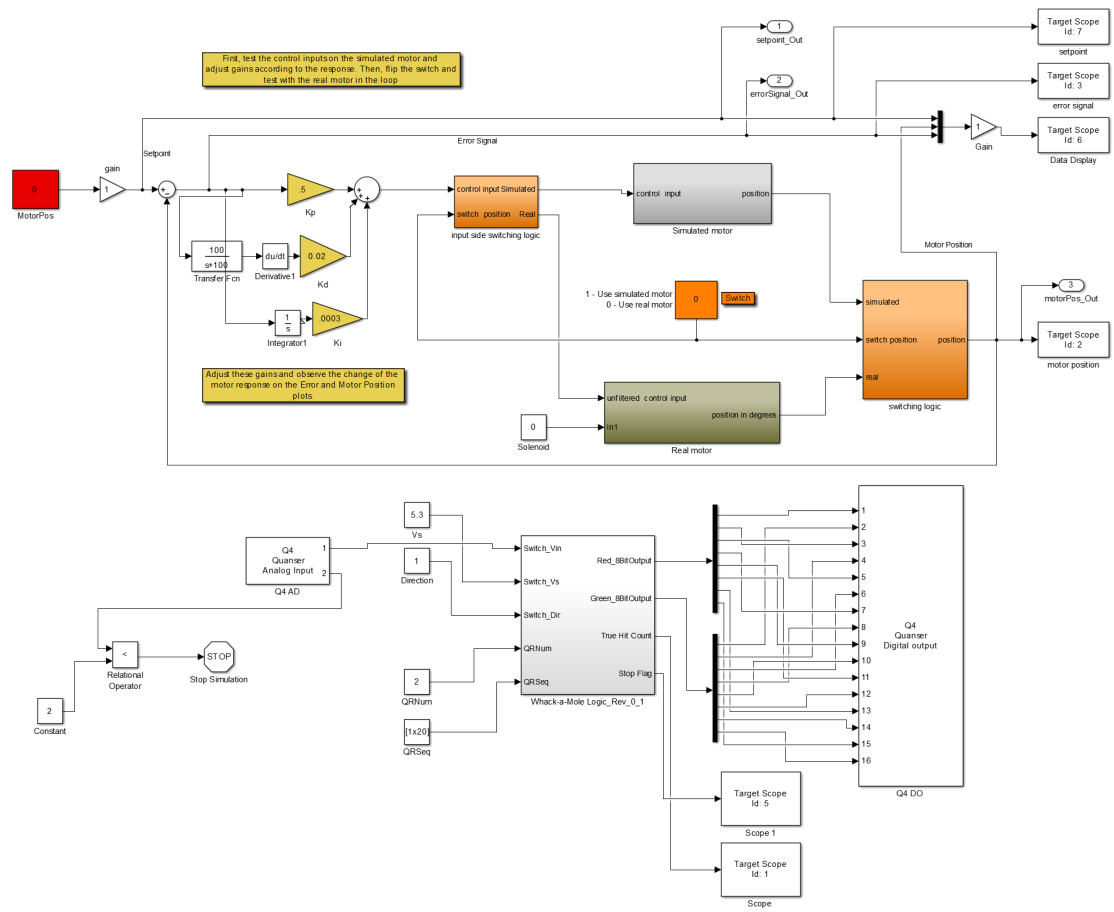


Figure 11: Simulink Whack-a-mole circuit and model.

Costs:

Item	Quantity[#]	Cost [$]
LED's	30	9.75
Push Buttons	10	12.45
Circuit Components Kit	2	55.00
Resistors	20	6.00
Wire	90 ft	8.99
Wood	mixed	12.00
Screws	1sm box	6.00
Solenoid	2	23.00
Emergency Stop Button	1	6.00
Slider Assembly Kit	1	90.00
Random Hardware Pieces	lot	20.00
Other		30.00
Total		279.19

Analysis of Final Prototype Performance

Overall the unit succeeded in meeting the customer requirements. The unit was able to perform nearly all the desired tasks, but failed to perform perfectly.

Construction Flaws

There were several problems mounting the motor vertically. We attempted to use several different motor mounts, but none of which worked perfectly well. Instead we were forced to construct a housing that kept the unit stable and relieved it of vibration.
The design held a belt in place with two pulleys. One pulley was attached to the motor by the use of a coupled shaft. Because of the self-made housing the distance was greater than anticipated and created a very small amount of space within the coupling. Checking and retightening frequently resolved this problem.
The solenoid that was used was not the original selection and the mount designed for the original did not fit perfectly. We modified the unit ultimately using zip-ties to keep the mechanism attached.

Electrical Flaws

The LEDs were very dim to the eye. This created some problems with the camera reading in all the centroids and delayed the entire process. The wiring was found to have a short, but was cleaned up and re-soldered to provide optimal circuitry.

Programming Flaws

Constraints in the programming restricted the amount of time the robot would play each round, which resulted in our demonstration leaving at least one light. We had problems with the signal error and only late in the project did we implement a cascading feedback function to alleviate the angle problems we were incurring. Overall, we could improve timing for this project.

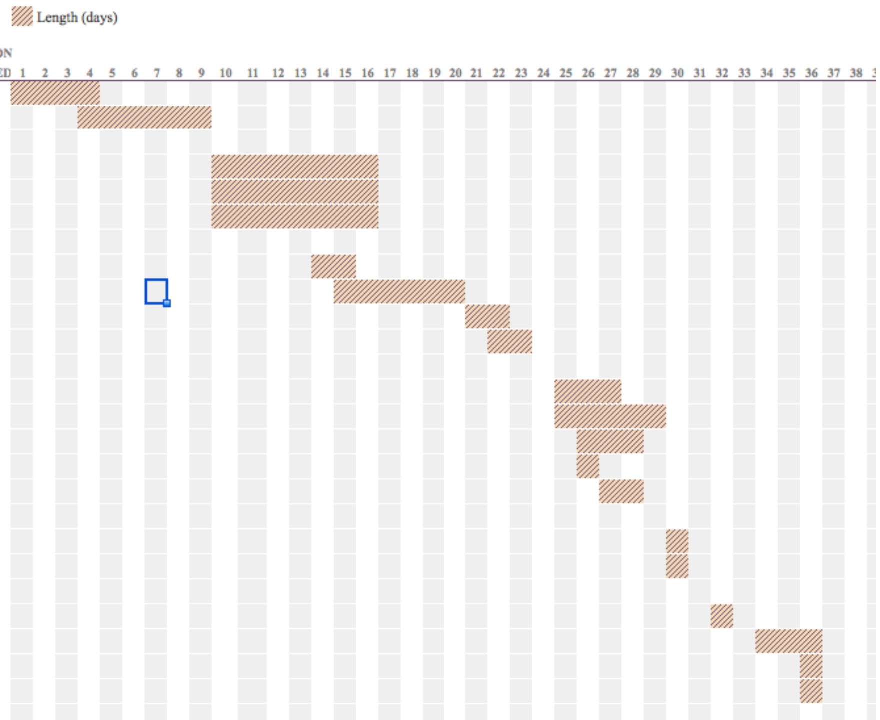
Customer Requirements

Topic	Target	Actual	Comments
AC Power	120 V	120 V	Used a DC supply & Amplifier additionally
Gameplay	Autonomous	Autonomous	Through mechanical, electrical and computing efforts the unit ran autonomously
Cost	< \$300	~\$279	One solenoid blew, bought several extra parts for "just in case"
e-component built	All circuitry	All circuitry	Using various components the system was built with basic circuitry elements
Fast Solve Time	Top 10%	Unsure	Never saw the final results, but expect it went well
Game board	Specifications	Specifications	Modeled after given specs and built precisely
Low Noise	<85dB peak	<85dB peak	Extremely quiet unit
Reliable	Works consistently	Works consistently	Soundly built with quality components
Safe	Not harmful	Not harmful	No dangerous pieces, e-stop installed

Gantt Chart

Gantt Chart Final Project

ACTIVITY	START	DURATION ALLOTTED
Design layout of board	1	4
Order track/arm parts	4	6
~Milestone 1~ 10/30		
Order E-Stop button	10	7
Order LEDs	10	7
Order Silenoid	10	7
~Milestone 2~ 11/13		
Build housing for board/camera	14	2
Solder PCB	15	6
Solder push-buttons	21	2
Solder LEDs	22	2
~Milestone 3~ 11/20		
Compile previous project's code	25	3
Code GUI	25	5
Code QR code	26	3
Figure out motor positions	26	1
Fix gain tuning	27	2
~IEEE Demo~ 12/2		
Run tests on system	30	1
Troubleshoot code	30	1
~Competition~ 12/4		
Final tests before demo	32	1
Divvy out Final Report parts	34	3
Submit parts	36	1
Edit final draft	36	1



ECE 4950 – Integrated System Design I
Project 6 Customer Requirements

<u>Project 6: Whack-a-Mole / Shocker Evaluation</u>

Group Name and Members: ____

Score	Pts		ABET Outcomes
	1	General Report Format - Professional Looking Document/Preparation (whole document) a) Fonts, margins (11pt, times new roman, single spaced. 1" margins on all sides). Follows the page limitations below. b) Spelling and grammar are correct c) Layout of pictures – all figures need captions and must be referenced in text d) References. Use IEEE reference format. *e) Report components are included in your website* *f) Grading sheet* **<u>Page 1: Title, Group Name, Group Members, and Date</u>** **<u>Customer Requirements</u>** *a)* Description of what the Customer wants	g
	2	**<u>Page 2: Engineering Requirements</u>** (<1 page) Bulleted list of Engineering Requirements	g
	20	**<u>Pages: 3-7: Design Details</u>** (<5 pages) Describe a system that can be built including System Architecture and System Integration <u>(See appendix for guidance)</u>. Do not include data sheets or software code.	k
	5	**<u>Page 8: Analysis of Final Prototype Performance</u>** (<1 page) 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.	c
	2	**<u>Page 9: Project Schedule/Gantt Chart</u>** (<1 page) 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
		Page 10 This grading sheet is included as the final page.	
	60	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)	c
	10	Rating by reviewers during competition.	k