ECE 4950 Project 6 (December 9, 2015)
Whack-a-Mole Evaluation
Team W8: Bode, Bode, Bode, Bode Plotting Everywhere

Ryan Barker
Nick Beulick
Peter Gallagher
Willie Brown
Rishabh Agarwal

I. Customer Requirements

- Autonomously play game according to “Whack-a-mole Description and Rules”.
  - For consistent timing of each round, start time will be calculated when the Game Master says “start”.
  - Any interpretation and clarification of the rules should be done through the course instructor (not the TAs).
  - No manual intervention once game is started
- Whack-a-mole game boards must be made based on the provide schematic and specifications. Any support structure must be done by the team.
- Cost* < $300 for parts beyond the cost of the xPC Workstation
- Use xPC workstation including Q4 board and Techron amplifier (if needed)
- Power Source for custom electronics (beyond the standard computers and peripherals such as a USB camera):
  - AC power - You may use an AC to DC converter
  - No batteries.
- Reliable, Durable, Safe
  - Low noise. Must be less than 85dB peak. Sound pressure level will be measured using “Decibel 10th” software version 3.7.0 from Skypaw running on an iPhone 4S. Default factory settings will be used. Measurement will be taken 2 feet from the front of the workstation.
  - Guarding as necessary to protect users.
  - A hardware emergency stop button to deactivate the system.
- Fast times for solving
- Easy to use, user friendly.
- Electric/electronic circuits built from basic components, e.g. resistor, inductor, capacitor, diode, transformer
- Runs autonomously during puzzle solving mode
- Must have at least one degree of freedom that has closed-loop position control where the loop is closed through the xPC Target.

II. Engineering Requirements

- LED circuit built properly
  - All 8 sections of the game board must be wired correctly
  - LED wiring must be done in a neat and organized manner
  - All LED's must be soldered on PCB board
  - Must only light up one LED at a time, as determined by the QR code

- Push button circuit built properly
  - All push button wiring must be soldered on PCB board
  - Must relay signal to software once a button is pushed
  - Push button wiring must be done in a neat and organized manner

- GUI must be user friendly
  - GUI must be formatted in a manner such that it is easy for any user to navigate
  - GUI functions to read QR code, seed random LED data, start program, and time game

- Read QR Codes
  - Have software identify the QR code from the other areas of the captured image
  - Relay QR Code data back to software
  - Software must analyze and sort data transmitted by QR code

- Locate active LEDs
  - Activate LEDs based on QR code information
  - Send image information back to software
  - Image spans the breadth of the game board
  - Cart travels across board and hits push button within time constraints

- Differentiate LED colors
  - Identify specific colors of LEDs from others
  - Relay LED color information to software

- Operate accurately and quickly
  - Software able to identify QR patterns and LED lights/colors effectively
  - Processing streamlined and flow of data concise for speed
  - Wiring is simplified and effective
  - Motor turns gear correct amount, leaving cart in appropriate position
  - Actuator hits push button once cart is stationary
  - Once button is pushed, light turns off and cart moves back to start position

III. System Architecture of Design

Figure 1 shows the Simulink system architecture for the final “Whack-a-Mole” system design, denoted the “Molehill”. The Molehill is comprised of five different subsystem circuits in three different parts: A Motor control circuit, a game logic circuit, a timing circuit, a solenoid circuit, and an emergency stop circuit. The overall Molehill circuit contained six inputs. Each circuit was designed and tested modularly to ensure easy integration into the overall final system.

The first main subsystem in the Simulink circuit was the game logic circuit. The brain of this circuitry was the Molehill block provided by the lab assistants, which used two random signal (capable of being seeded by the overall system GUI via a pushbutton) to pseudo-randomly light LEDs. The time duration LEDs lit were influenced by “QRNum”, which was an input for the round number contained in the current QR code being interpreted by the system. “Switch VIN” was connected to the output of the pushbutton subsystem so the Molehill could determine if a pushbutton was pressed to light the next LED. The pushbutton subsystem output was also displayed on a numeric target scope to allow for easy monitoring. “Switch VS” was equal to the voltage supplied to the pushbuttons, so the circuit could verify the correct voltage across each pushbutton. “QRSeq” was used in “No Tricks Rounds” (Denoted “QRNum” = 1 where the robot needs to hit the corresponding pushbutton for every LED it sees) to determine the lighting sequence. “Red_8BitOutput” and “Green_8BitOutput” were sent to the LED lighting circuit to control the game board lighting.

The second main subsystem in the circuit was the emergency stop system. This worked utilizing a latching, normally open switch and by connecting the input side of the switch to a five volt power supply. The output side of the switch was read from the Q4 board by the Simulink circuit (Analog Input 4) and compared to a two volt signal to eliminate the effects of noise in the circuit. Once the pushbutton was pressed, the circuit completed, and a high signal was sent to a STOP block, terminating the Simulink session.

Other subsystems included in the file were motor control, solenoid, and timing subsystems. The motor control circuit was a direct copy of the Project 5 motor control circuit, which used a PID loopback controller and a low pass filter at the derivative filter to eliminate physical noise from the subsystem. “MotorPos” was the desired position of the motor in degrees. The timing circuit was a simple clock block connected to a numeric target scope and system output so the final GUI could display the time elapsed. The solenoid subsystem connected the “Solenoid” constant block to the input of amplifier channel two, which was used to drive the solenoid. If “Solenoid” was changed to one volt, the subsystem caused the solenoid arm to fire.

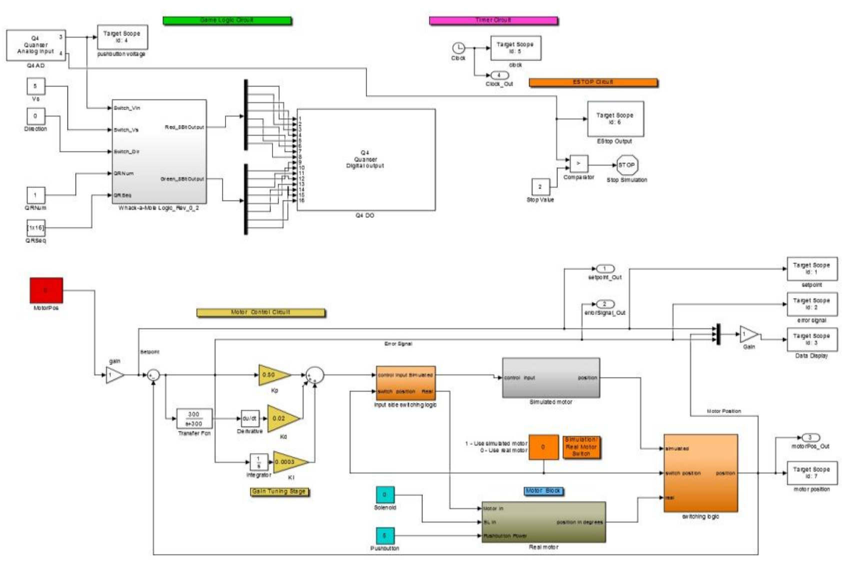


Figure 1. Overall System Architecture.

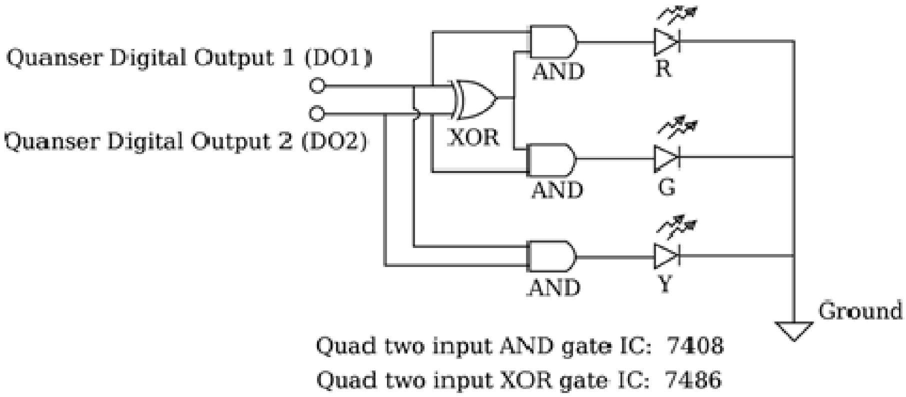
IV. Electrical Components of Design

The electronic components list we used for our final design can be seen in Figure 2 below. Three Adafruit Perma-Proto Full Sized Breadboard perforation boards were used to wire the LED circuits as well as the pushbutton circuit. Eight of each desired color LED (red, green, and yellow) were used to create the game board along with eight pushbuttons. The other electrical components to be outlined in this section are the Amico emergency off button, the Tohoku motor to push the cart, and finally the push-pull solenoid that will actuate the pushbuttons.

Component	Amount
Push-pull Solenoid	1
16mm Momentary Pushbuttons	8
10kΩ Resistors	8
Red LEDs	8
Green LEDs	8
Yellow LEDs	8
PCBs	3
7408 AND Gates	7
7486 XOR Gates	2
Emergency Stop Button	1
Tohoku Motor	1
Q4 Quanser Board	1
Amplifier	1
Wiring Connecting All Areas	> 100

Table 2. Electrical Component List.

Two perforation boards were used to permanently connect the LED circuits to the game board. Three 7408 AND gates and one 7486 XOR gate were going to be placed on both perforation boards, but four AND gates ended up being soldered to one board because of another faulty AND gate. The logic for each section of the board matched the wiring in Figure 3 exactly. DO1 for each circuit were wired to the Quanser Q4 board digital outputs one through eight. DO2 for each circuit were wired to the Quanser Q4 board digital outputs eight through sixteen. The LED circuit was powered by global power and ground buses, connected to a DC power supply. The ground lead of this supply was tied to the Q4's ground to normalize the grounding between the two systems. Each LED circuit was tested as it was completed to ensure it worked properly before building the next. This was done with a very simple Simulink circuit which connected constant blocks to each of the relevant digital outputs. Once the perforation boards were completed, the LEDs and boards were placed into the physical system and soldered together to make the connections permanent. Two LED inputs (section two yellow and section eight green) ended up also incorrectly lighting red LEDs as a result of cold solder joints snapping during this process, but the errors were accounted for in software and did not affect the overall system's functionality.

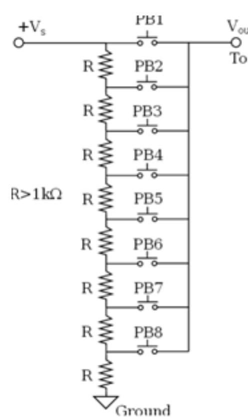


Truth Table

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 3. Single Game Board Section Circuit.

The third perforation board was used for the pushbutton circuit. Using the wiring diagram in Figure 4, eight 10kΩ resistors were placed in a series pulldown network and pushbuttons were connected to each junction point in the network. This way, the voltages at the output of each pushbutton were eighth multiples of the input voltage. The input of this circuit was connected to the Q4 board's analog output three and the output was read from the Q4 board's analog input three. The circuit was tested by sending five volts through Simulink, and observing zero volts across the circuit normally and the output voltage change by eighths as each pushbutton was pressed. Once these values were correct along the circuit, the pushbuttons were soldered to the game board. This marked the completion of the game board circuitry, so the LEDs and pushbuttons on the board itself were glued down to ensure they did not move while the robot was in motion.



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

Figure 4. Pushbutton Circuit.

The emergency stop button circuit was implemented as a safety measure to shut off the system completely in case of potential damage to the user or equipment. The button used was a normally open latching switch from Amico. Its input side was connected to a five volt DC power supply and its output side was read from analog input four of the Q4 board. This caused the system to operate normally while the button was not pressed, but stopped the system when the button latched down. To test the emergency stop circuit, we set up a circuit that continuously fired a solenoid, executed the circuit, pressed the emergency stop button, and observed the solenoid stopping. The emergency stop was placed on the front of our design for easy accessibility.

The Tohoku motor and Solenoid used were each connected to amplifier channels that were driven by analog inputs one and two of the Q4 board. The motor circuitry was identical to what was used in projects four and five, and the solenoid did not require a driver circuit, so it was connected directly to amplifier channel two. Small subsystem Simulink circuits and MATLAB code were set up to test the motor and solenoid by continuously moving the motor to specific locations (and by consequence, the tread used in the design) and continuously firing the solenoid. The motor gains were re-tuned once it was correctly mounted to the tread to optimize its performance and minimize its error. The final gains used were $K_p = 0.50$, $K_d = 0.02$, and $K_i = 0.0003$. A low pass filter with cutoff frequency 300 was implemented in front of the derivative gain to reduce the overall noise from the motor.

V. Mechanical Components of Design

The mechanical components used for the final design can be seen in Figure 5 below. The core structure of our design was wood based and was used to mount the various core components such as the motor, gears, and guide rails. Figure 6 contains a picture of the final design. The idea behind the structure was to have our motor mounted in the back of the design so that the movement of the cart could be controlled using our tread and gears. We called this our "laser printer" style design. The gears were attached to small wood pieces above the motor and on the opposite side of the back of the structure in order to allow the tread to have a tight and continuous looping path. Attached at the top of the design are two metal rods, which were inserted as a way to stabilize the arm that carried our solenoid, which was given motion from the tread in the back of the design. Once the software was synced up to the hardware, we found that the arm which connected the tread to the cart needed to be stabilized during motion. This was because once the software told the motor to turn, the tread would tend to turn the arm and cart diagonally, instead of moving in a strictly horizontal motion. These two metal rods were inserted through washers, which were attached to the "arm", which stabilized the motion of the cart. The motion was still restricted because of increasing friction at certain points on the track, but these differences were accounted for in the software. All dimensions match what is described in the manual, and our game playing system is 8" x 24".

Components	Amount
Motor	1
Gears	3
Tank Tread	1
Metal Rod	2
Wood Cuts	18
Web Cam	1
Cart Kit	1

Figure 5. Hardware Components List.

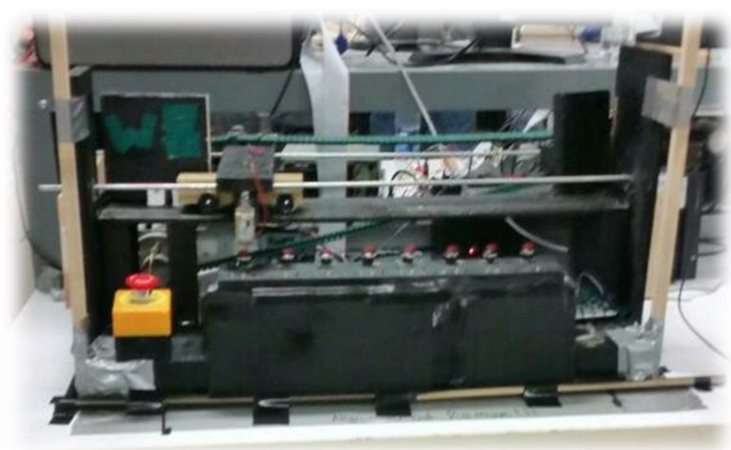


Figure 6. Image of final design.

VI. Software Components of Design

Figure 7. MATLAB Game Logic Flow Chart.

All game logic for the Whack-A-Mole robot was implemented via MATLAB and is shown in Figure 7. It centered on computer vision, motor control, and integration with Simulink and XPC. To increase the speed and performance of the code used, the only data structures employed were the handles structure in the GUI code, which stored information related to the QR code round number, QR code sequence, QR sequence size, and handles for the relevant dynamic text boxes to be updated during the game code. The only sizable variable that the code stored in general was an initial background image of the system, which was key to the image processing. All foreground images were taken as they were needed to optimize efficiency. Additionally, the system was programmed for the motor to start and return to section zero of the game board after striking each LED. This was done to reduce the impact of setpoint error on the system's motor, allowing the solenoid to strike the center of each pushbutton every time it moved to a different board section. The image processing code utilized background subtraction in RGB and regionprops to identify the centroid of any lit LEDs on the game board, threshold the centroid's x component against the midpoints of the known LED centroid locations, and return the color and location of any lit LEDs. This method meant the camera could not move during execution, but achieved substantially faster performance than traditional background subtraction and color processing (the execution time was 1.5 seconds faster). The overall software package was tested with several sample QR codes on blackboard and generated on the web. We ran over 500 executions with at least 15 different QR codes and refined the system until its successful "mole-whacking" rate was over ninety-five percent (Typically, this meant missing two or less moles during game execution.)

VII. Budgeting and List of Materials

Component	Quantity	Unit Cost	Purchased From?	Ordered?
Push-Pull Solenoid	1	$14.95	https://www.adafruit.com/products/413	Yes
16mm Momentary Pushbuttons	8	$0.95	http://www.adafruit.com/products/1503	Yes
Red Mouser LEDs	8	$0.50	ıologies/HLMP-1700/?qs=sGAEpiMZZMtm	Yes
Green Mouser LEDs	8	$0.50	nologies/HLMP-1790/?qs=sGAEpiMZZMtn	Yes
Yellow Mouser LEDs	8	$0.50	ıologies/HLMP-1719/?qs=sGAEpiMZZMtm	Yes
PCB 3-Pack	1	$19.95	http://www.adafruit.com/products/590	Yes
Pinewood Derby Car	1	$9.39	ım_medium=cpc&utm_source=googlepla&	Yes
Vex Tank Tread Kit	1	$38.99	/www.robotshop.com/en/vex-tank-tread-kit	Yes
Amico Red Sign Mushroom Emergency Stop Push Button Switch	1	$6.65	ı_sim_328_1?ie=UTF8&dpID=41wxITX7B	Yes
Female/Female Jumper Wire 40-Pack	2	$3.95	https://www.adafruit.com/products/266	Yes
Male/Male Jumper Wire 40-Pack	5	$3.95	https://www.adafruit.com/products/758	Yes
Female/Male Jumper Wire 40-Pack	3	$3.95	https://www.adafruit.com/products/826	Yes
Large Recoil Automatic Cord Winder	1	$9.95	nder/large-recoil-cord-winder-untangles-h	Yes
Screws	8	$0.25	Ace Hardware	N/A
Steel Support Rods	2	$3.35	Lowe's	N/A
Duct Tape	1	$3.88	Lowe's	N/A
Gorilla Glue	1	$5.98	Lowe's	N/A
Black Spray Paint	2	$4.98	Lowe's	N/A
12 ct Screws	1	$1.98	Lowe's	N/A
10 ct Screws	1	$1.98	Lowe's	N/A
8 ct Screws	1	$1.94	Lowe's	N/A
Coupling Fittings (used as weights)	2	$2.78	Lowe's	N/A
Poplar Board	1	$8.33	Lowe's	N/A
Hinges	5	$1.97	Lowe's	N/A

Total:	$217.14

Figure 8. Whack-A-Mole System Budget.

Figure 8 shows the budget of the final design. This was created in a Google sheet and updated as components were bought. The total section of the sheet live-updated, which allowed us to ensure we met our three hundred dollar budget. All components used in this project are easily ordered from Adafruit.com, Ace Hardware, or Lowe's Hardware.

VIII. Analysis of Final Prototype Performance

Customer Requirements	How Design Meets Requirements
Autonomously play game according to "Whack-a-mole Description and Rules"	Our design autonomously plays the "Whack-a-mole" game as described in the rules and specifications for each of the 4 rounds.
Whack-a-mole game boards must be made based on the provide schematic and specifications. Any support structure must be done by the team.	Game board meets dimensional requirements. Tread system was assembled by team and all supports were custom made by team.
Cost < $300	Total system cost: $217.14
Use xPC workstation including Q4 board and Techron amplifier (if needed)	The xPC workstation, Quanser Q4 board, and Techron amplifier were all incorporated into the design of our project.
Power Source for custom electronics (beyond the standard computers and peripherals)	The provided DC 5 volt power supply was used.
Reliable, Durable, Safe	System is reliable, durable and safe. Extra wood supports were added to components of the project to ensure stability, along with components being screwed into wood with longer screws to provide additional support. All electrical components were covered and insulated.
Fast times for solving	Round 1 times were all in the mid-30 sec range
Easy to use, user friendly.	GUI is very user-friendly, and is only component that requires user interaction.
Electric/electronic circuits built from basic components, e.g. resistor, inductor, capacitor, diode, transformer	All circuits were soldered onto perforation board. Individual wires/pushbuttons were soldered, and the rest of the circuitry consisted of integrated circuits and resistors from the ECE lab kit.
Runs autonomously during puzzle solving mode	Robot plays game autonomously
Must have at least one degree of freedom that has closed-loop position control where the loop is closed through the xPC Target.	X-direction motion used in the tread/cart system (one degree of freedom) is run by using a Simulink circuit which was controlled through the xPC Target (closed-loop position control).

Figure 9. Requirements Table.

IX. Project Schedule / Gantt Chart

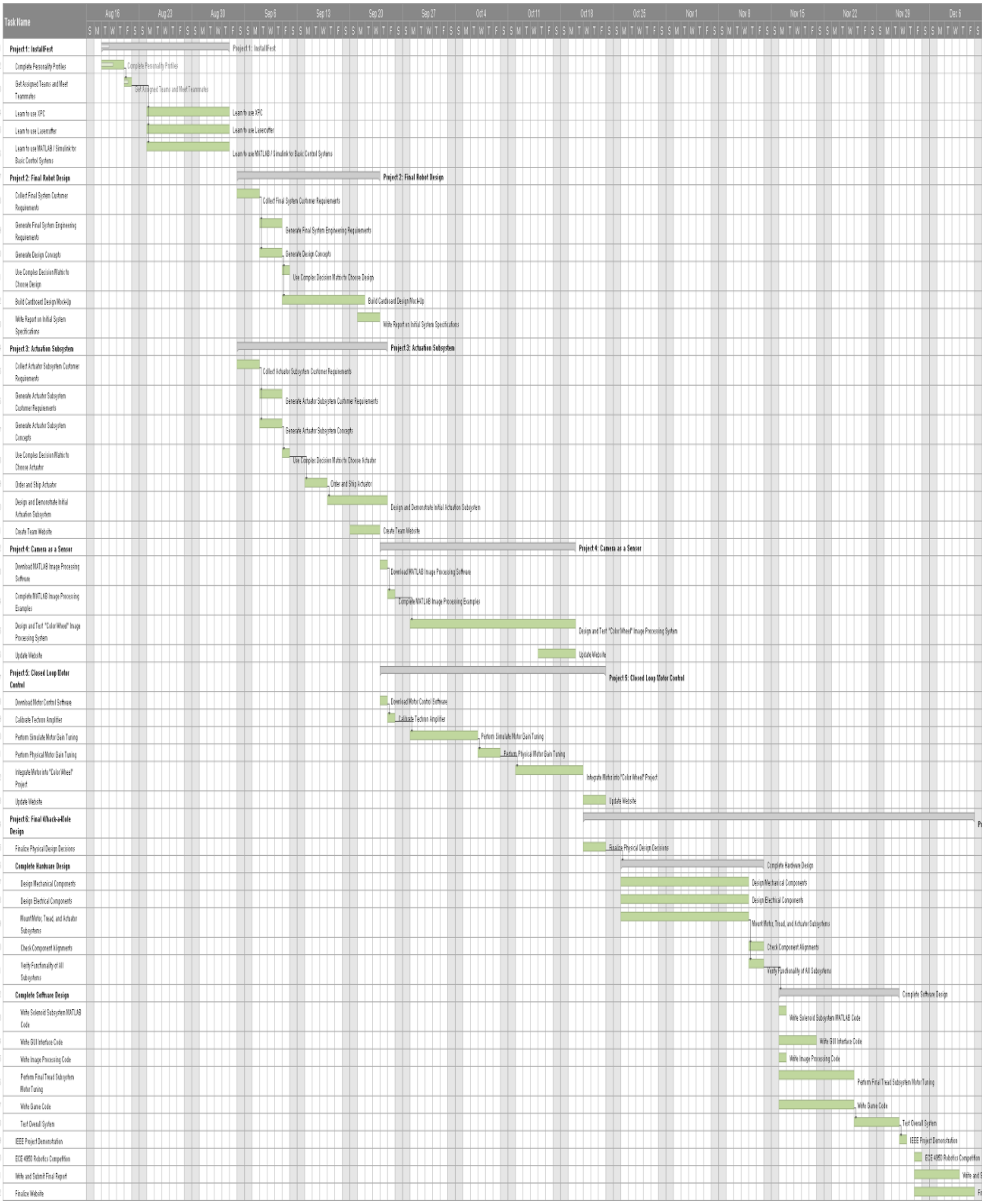


Figure 10. Gantt Chart.

Our website contains a much more viewable variant of this photo (http://people.clemson.edu/~phg/home.html).

X. References

[1] A. Kapadia. ECE 4950 - Integrated System Design [Online]. Available: http://people.clemson.edu/~akapadi/ece4950_references.html. Cited: 12/06/2015.

ECE 4950 – Integrated System Design I
Project 6 Customer Requirements

Project 6: Whack-a-Mole / Shocker Evaluation

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* **Page 1: Title, Group Name, Group Members, and Date** **Customer Requirements** *a)* Description of what the Customer wants	g
	2	**Page 2: Engineering Requirements** (<1 page) Bulleted list of Engineering Requirements	g
	20	**Pages: 3-7: Design Details** (<5 pages) Describe a system that can be built including System Architecture and System Integration (See appendix for guidance). Do not include data sheets or software code.	k
	5	**Page 8: Analysis of Final Prototype Performance** (<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	**Page 9: Project Schedule/Gantt Chart** (<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