From : University of Houston

4800 Calhoun Rd.

Houston, Texas 77004

Date: September 16, 2016

Subject: Burns & McDonnell – PLC Microgrid Project: Progress Report

To: Cary Gallaway and Austin Arnett

Burns & McDonnell

1700 West Loop South Suite 1500

Houston, Texas 77027

Dear Mr. Gallaway and Mr. Arnett:

Enclosed you will find our final report detailing the finished product for our Burns and McDonnell - PLC Microgrid Project. We will revisit our target objective along with the specifications initially given to us, and describe how we carried out what you ask of this project.

We are very proud of our final product and we think that this document will give you great insight into the work put in to accomplish it. Over the course of this year we have taken all of your concerns and specifications and we feel that we have met and exceeded all of them.

We hope this document satisfies any questions you may have on the finished product. We look forward to delivering the model and letting you see it in person. Thank you for the great opportunity to build this model for Burns & McDonnell and your support throughout the project.

Sincerely,

Patrick Gros

John Nemec

Larry Hernandez

Burns & McDonnell –

PLC Microgrid Project



**2016 University of Houston Capstone Project**

Patrick Gros, Larry hernandez, john nemec

Sponsors: Austin Arnett, cary gallaway

fACILiTATOR: Dr. Steven Pei

**Abstract**

Burns and McDonnell’s recent involvement in microgrid projects has highlighted the new challenges inherent in the PLC's role in the automation of a power system. This project aims to promote the understanding of the PLC’s role in a microgrid environment and it’s capability to boost the systems reliability and improve upon economic conditions. The deliverables of this project include a fully functional scaled microgrid model composed of various loads, disconnect switches, and distributed generation sources. This provides the user with a physical demonstration of the PLC’s interaction in a microgrid. In addition, a control panel was installed to allow the user to impose various conditions on the microgrid and a display screen that illustrates the microgrids reaction to these impositions. An Arduino Mega was employed to play the role of the system and execute the actions on the microgrid. A Raspberry Pi 3 was installed to act as the systems PLC by taking in data from the Arduino (the system) and administer the appropriate action as a result of the systems current state. Due to hardware limitations in the microcontrollers, recommendations are made to enhance data resolution display screen. Additional recommendations are made towards easing the mobility of the model.

**Intro, Background, and Goal**

This document is a report detailing the final status of the Burns and McDonnell - PLC Microgrid Project. It contains the project’s overview diagram, target objective/goal analysis, engineering specifications and constraints, the test plan for our completed objectives (including a Gantt chart depicting the project schedule), a progress description reporting the objectives accomplished, and finally the project’s budget summary.

The overall objective of this project was to present to Burns and McDonnell a functional scaled model of a microgrid power system that can demonstrate to a user how a PLC (programmable logic controller) functions in a microgrid environment and how it is capable of boosting power dependability in the system and capitalize on economic conditions.

In this project, we aimed to design a controller composed of various switches that will impose predetermined conditions (established by Burns and McDonnell) on the microgrid power system, then program a microcontroller/PLC (Raspberry Pi) to respond to these conditions with the mindset of maximizing the reliability and economic situation of the system. The data and actions of the system are to be presented on a video monitor screen for the user to visualize. Effectively, this project aims to mimic a real-world microgrid power system and will be employed to introduce and educate PLC/microgrid integration.

**Overview Diagram**

The figure below shows the overview diagram for the microgrid model.

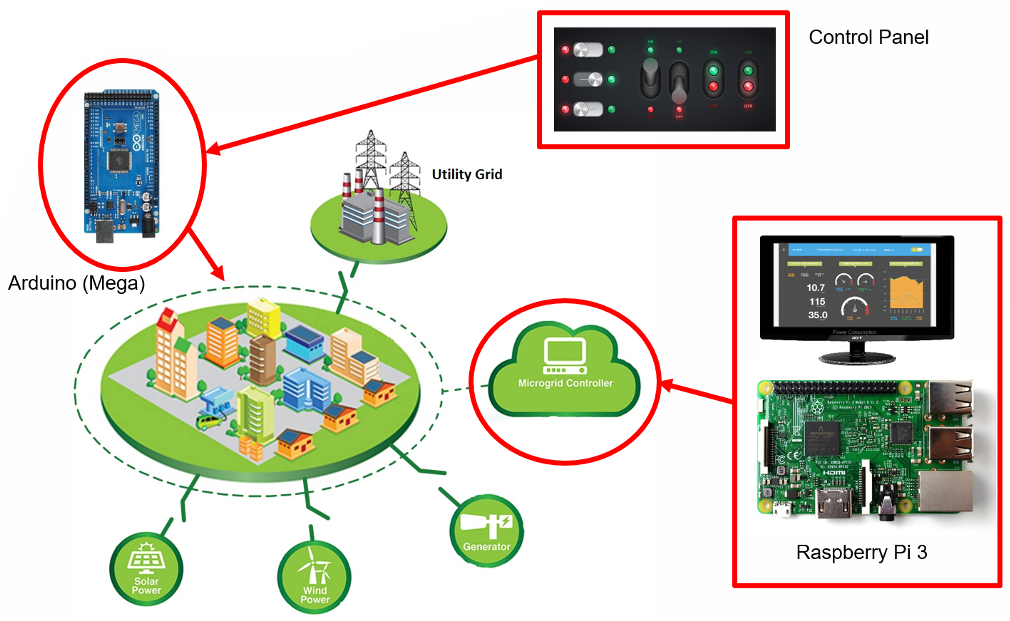


Figure 1 – Overview diagram for the microgrid model

As can be seen in the figure, the microgrid is enclosed by the dashed line and the various distributed generation sources surrounding it. The microgrid will be actuated by an Arduino Mega, with the digital I/O pins controlling the LEDs and servo motors on the model. The microgrid is controlled by a PLC, shown as a Microgrid Controller, which has the option to switch to various distributed power sources. We will use a Raspberry Pi 3 for the controller, which will control the Arduino Mega, and thus the LEDs and servos. There will also be a Control Panel, which will be connected to the Arduino, and a display connected Raspberry Pi 3.

**Target Objective and Goal Analysis**

The Figure below shows a graphical representation of the goals that were completed to successfully deliver the target objective, which is in the box furthest to the right.

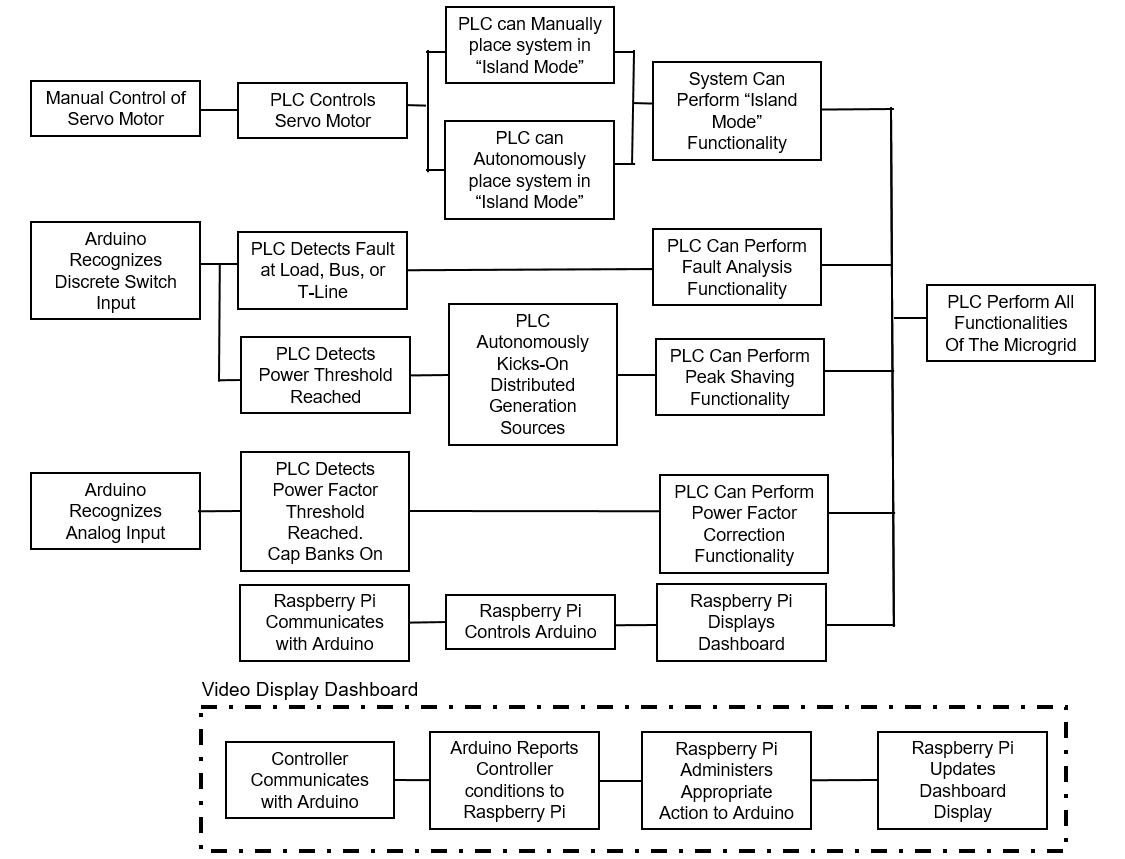


Figure 2 – Goal Analysis

As shown in the figure, the goal analysis diagram is broken into different horizontal “tracks”, each ending in a specific function that must be met to accomplish the target objective. It will be our strategy to have each team member start down a different track and accomplish the objectives for each function. All of which serve the purpose of meeting the target objective: “PLC to perform all functionalities of the microgrid.”

**Engineering Specifications and Constraints**

The specifications for the microgrid model have been developed out of several conversations with the engineers at Burns & McDonnell. In these discussions, several project specifications were established, and are as follows:

* Microgrid is to have the ability to go into Island Mode in the event of a utility outage or as economic conditions dictate.
* Microgrid is to autonomously execute VAR compensation
* Microgrid is to react to fault situations, as well as load shedding
* Microgrid is to implement peak shaving
* Microgrid is to have Graphical User Interface (GUI) and to be user friendly for easy comprehension.

After taking in the specifications and coming up the analysis documentations, several constraints were identified and are as follows:

* Model size and weight: While it was not a key specification laid out by the engineers at Burns & McDonnell, there was mention of the model be mobile and possibly have the ability to hang on the wall. Thus, the size and weight will have to be taken under consideration as the project develops.
* Arduino limitations: The microcontroller that is under consideration is the Arduino Mega. There are certain limitations that must be accounted for, such as memory size, processor speed, number of available pins, digital I/O voltages, and max current output, to ensure the Arduino does not burn out. The Arduino must be able to overcome these limitations and perform flawlessly.
* Hardware constraints: The Arduino is going to interact with hardware that will be placed on the microgrid model. Thus, the hardware specifics will need to be taken under consideration. Items such as relays, servo motors, potentiometers, LED displays, and other electronic hardware must operate in such a way that the Arduino can have precise control at all times.

**Project Schedule and Test Plan:**

The following Gantt chart illustrates: the objectives to be completed this semester, the team member(s) assigned to that objective, and the date the completion date of that objective. All objectives have been completed and this chart is only included to show the division of labor as well as the scheduled completion dates of the various phases of the project.

Figure 3 - Project Schedule

**Design**

Burns and McDonnell gave us a lot of autonomy to design the microgrid model and PLC controller. We spent a fair amount of time discussing what would be most effective and came up with design in the following drawing.

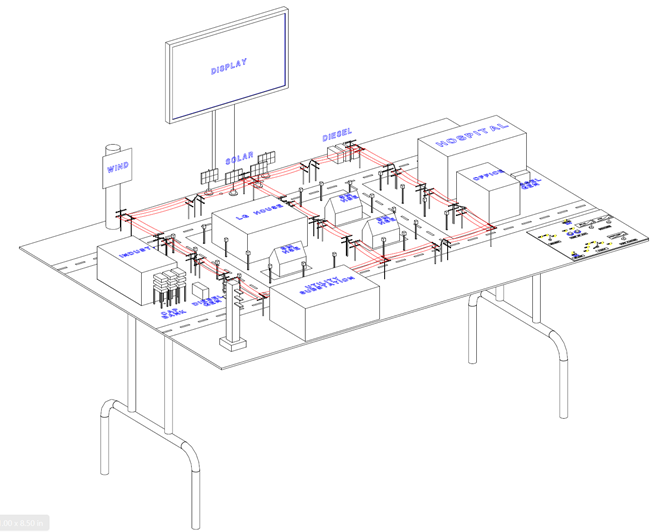


Figure – CAD version of microgrid model

As you can see in the above figure, we created a table of sorts with different model pieces, a control panel, and a monitor for display. We decided that this would be the most useful way to show how a PLC operates in a microgrid environment.

The model pieces are representative of the different loads in the microgrid. We have three distinct zones: industrial, residential, and office/hospital. Each of these loads are representative of the type of loads found in the real world and each has a distinct power demand.

The control panel is where the user will interface with the model. It gives the user an opportunity to impose various changes on the grid and see how the PLC responds. The user has the ability to change the time of year, choose day or night, fault out one of the loads add reactive VARs to the system, to change the intensity of the wind or sun for the distributed generations, and go into or out of island mode. With each one of these choices, the PLC must make decisions on how to best react to the imposed conditions.

To see how the PLC reacts, the user can look at the model itself or look at the monitor. This monitor is the HMI (human machine interface) and is where the PLC communicates with the user. It will tell the user how much power the loads are taking, how much generation is on the grid, and give indications of other aspects. It will show how the PLC is reacting in real time, and update the user on these reactions.

**Methodology**

After coming up with our model design we were tasked with deciding how to best implement all of the features. For the PLC/microgrid we used a Raspberry Pi 3 connected via USB to an Arduino Mega. This allowed us to use serial communication to control the input and output pins of the Arduino Mega. The program for the simulation was written in Python and runs on the Raspberry Pi 3. Within that program we are using libraries for PyGame and PyFirmata. PyGame is a graphical design library that allows the programmer to place shapes, text and color pixel by pixel. This library was used exclusively to construct our HMI that displays all pertinent information about the current conditions on the microgrid. PyFirmata is a library that allows the Raspberry Pi 3 to communicate serially with the Arduino. This library was crucial in the connection between our software simulation and the physical control of the model microgrid.

During the design process the team realized that the number of LEDs and servo motors being used would exceed that maximum current capabilities of the Arduino Mega. The team came up with the solution in the form of a 16-channel relay board that could be controlled by the arduino. Each relay is individually capable of handling a maximum of 30 Vdc and 10 Adc. This is well above the maximum possible current needed for the model to run safely and efficiently. Each relay is individually controlled by a pin from the Arduino Mega and tied to a pull down resistor to ensure that there are no incorrect voltages sent to the relays. The relay board itself is powered by its own 12V-2A power supply that provides power to the relays for switching. The relays themselves are also being powered by their own individual 12V-2A power supply that has the positive connection jumpered to each common terminal. Then each load is connected to the normally open contact of the relays. In our design we also took experience from our internships at Burns & McDonnell and decided to use a system of terminal blocks to aid in organization and ease of troubleshooting. Each incoming power supply first enters a terminal block and then leave to either another zone terminal block or, in the case of the relay board, directly to its device. The negative or ground connection of each power supply leaves the power terminal block and jumpers around the board making connections to each zone's respective terminal block. This design allowed us to take advantage of wiring the entire model in parallel. This means that if one load fails the rest of the model will continue to work while the load is being replaced.

The servo motors required a ground, positive, and signal connection. The positive and ground were supplied through another 12V-2A power supply. These leads were wired in parallel similarly to the loads. The positive and grounds were wired from terminal block to terminal block and then jumpered within each terminal block. This provided individual landing spots for each servo which again is for ease of troubleshooting and clarity of wiring. The signal wires for each servo were run directly from the Arduino to their respective servo motor. The servo motors were another reason for choosing to use the Arduino Mega. Each servo requires a PWM signal that would normally have to be programmed into the controller. The Arduino makes this easy by have a library that is able to mimic a PWM with an internal timer and a digital output pin. Therefore coding and using the servos with the Arduino is as simple as writing a degree output to the pin of that servo and letting the Arduino do the internal timing for the PWM.

On the input side of the microcontroller, the team decided to use an array on switches, push buttons, linear potentiometers, and rotary switches to allow the user complete control over the conditions on the microgrid. The top switch of the controller starts and stops the main code. After starting the code the user is prompted to use the first rotary switch to select either the Summer or Winter season. Next two push buttons allow the user to select either day or night. This takes the user through a simulation of a real microgrid at the given season and 12 hour time of day selected. Within this the user is allowed to use the rest of the controls to change conditions on the grid. The wind and sun intensity potentiometers allow the user to change the amount of sun or wind from the set number of the simulation. These changes are reflected on the HMI for the user to see. The user can also change the VARs on the grid using the reactance potentiometer. As this knob is turned up the amount of reactance on the grid increases and the power factor shown on the HMI starts to drop. As the power factor reaches 0.90 the PLC brings online a discrete cap bank that raises the power factor back to 0.98. The system is designed with three of these discrete cap banks representing the maximum amount of VAR compensation needed for this system. The HMI displays both the corrected and uncorrected power factor as well as turning on LEDs in front of each capacitor back as the are brought online. The island mode switch allows the user to see how the grid uses the distributed generation sources in the event that connecting to the utility is no longer viable. The final rotary switch allow the user to create a fault at a determined position in the grid and see how the PLC would correct or self-heal in order to maintain service to as many loads as possible. These faults include a fault on the industrial load, a fault on the residential load, a fault on the office load, and a fault on the bus. After selecting the fault on the rotary switch the user can press the push button below the rotary knob to actuate the fault simulation. All of these inputs are individually wired to their own input pin on the Arduino Mega.

**Results**

As previously discussed, we built a model microgrid with model pieces, a control panel, and a monitor for display. Below are some pictures of what we built and are the results of our hard work. First you will see some of the model pieces.



Figure – Residential model Piece



Figure – Hospital and Office building model pieces



Figure – Industrial model piece

As you can see in the pictures, the model pieces are high quality and make it easy see what kind of load you are looking at. Majority of the model pieces were built and painted by hand, with lots of care for the details to enhance the user experience. Also in the pictures you can see that we added street lights and switches in the power lines to give indications of what is happening on the grid.

Below is a screenshot of the HMI display for the user.

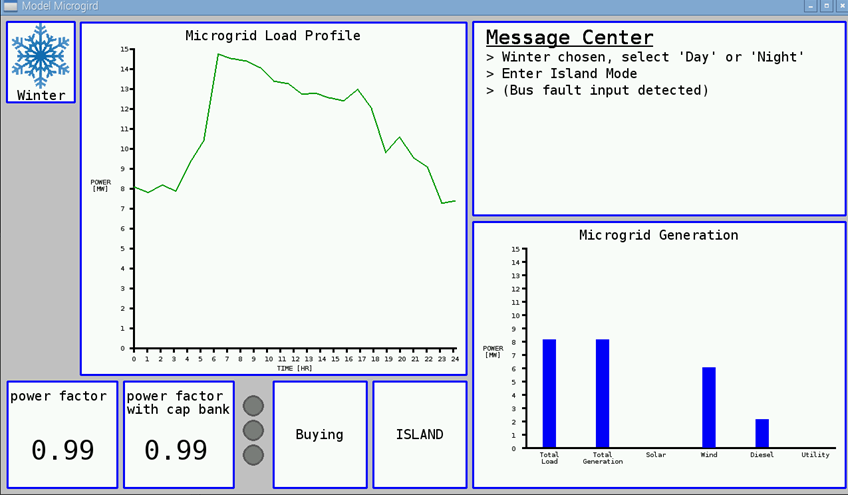


Figure – User display

As you can see, the user is given a lot of information about the status of the microgrid, such as the load's demand, the generation on the grid, the power factor, the island mode status, whether buying or selling from the utility, and even has a message center to output text messages.

Below is an image of the user controller.



Figure – Control Panel

It is easy to see how the user can adjust various elements on the microgrid model by flipping switches, adjusting rotary dials, pressing buttons, and turning potentiometers.

We feel like these results meet the specifications laid out at the beginning of the project. All the points requested have been touched upon and done in a way that is user friendly and appealing.

**Conclusion**

In conclusion, the delivered product has successfully accomplished the target objective, imposed by Burns and McDonnell, to promote the understanding of a PLC’s role in a microgrid environment. Burns and McDonnell’s list of specifications were thoroughly addressed and overall the client was exceeding pleased with the final product. This model will be employed by Burns and McDonnell to bring to future STEM (Science, Technology, Engineering, and Mathematics) events and to educate budding engineers on microgrid power systems. The project was completed on time and within the proposed budget of $14,074. It is our view that future improvements for this project should target increased data resolution to enhance the realism of the model, additional hardware to include additional data points on the system, a touch screen controller, and finally higher quality servo motors to increase the smoothness of the motors rotations.

**Recommendations**

While we were successful in building a model that demonstrates how a PLC operates in a microgrid environment, there are a few things that can be done to improve the model. One of these is to increase the resolution of the load data. Currently we are using 24 discrete data points for the 24 hours in the day. If we could increase these to be 96 data points to be every 15 minutes or 1440 data points for every minute, then the model would be that much better. Also more sources of data would be a good addition, to show system voltages, currents, and any other pertinent information. Better servo motors and a touch screen would be a nice touch, as well as making improvements to the ease of portability. Despite these recommendations, the model as it stands now is a very effective tool.

**Financial Summary**

The figure below shows an outline of cost of the entire project.

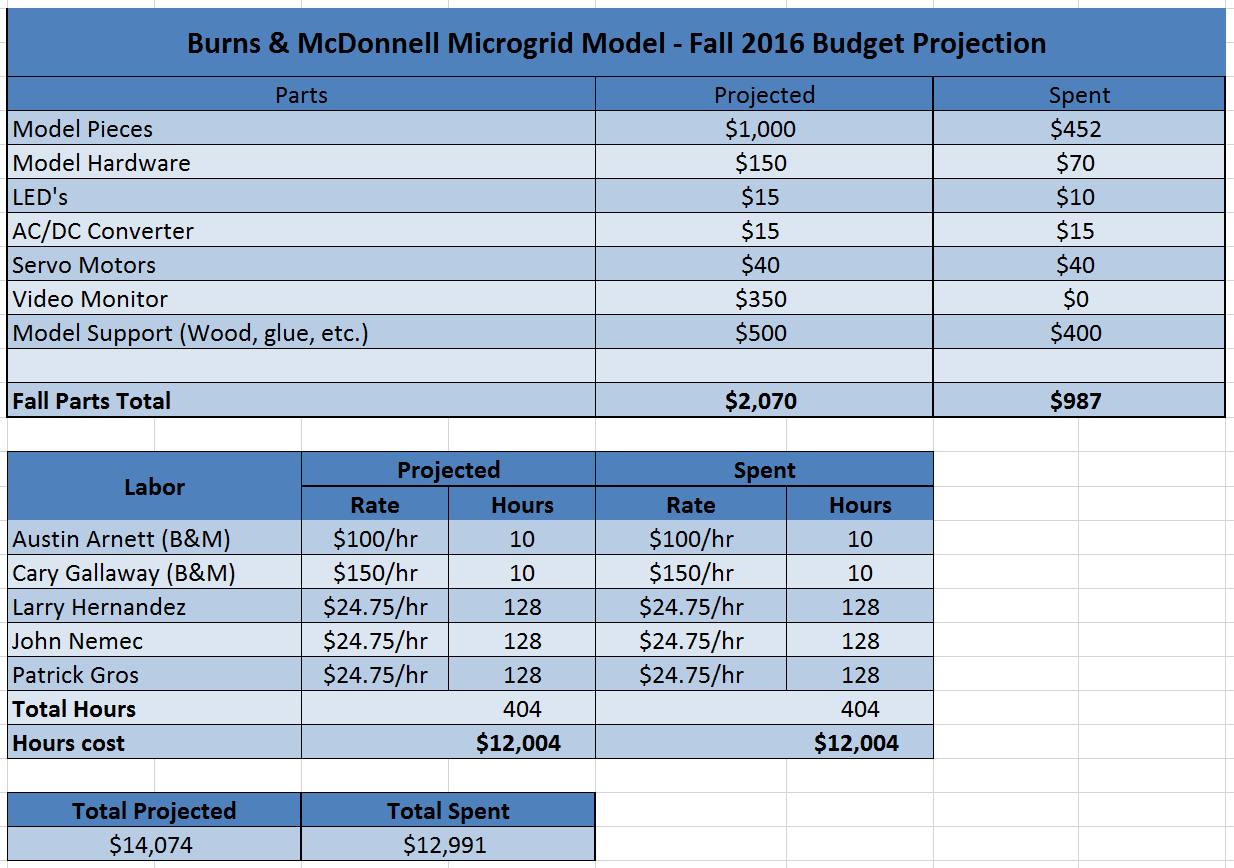


Figure 10 - Final Budget

One thing you will notice is that we came in under budget on this project by nearly $1000. This is primarily because we did not as much as we anticipated on model pieces. Also, we budgeted money for a monitor, but as of right now we have one donated from the office. These two categories account for our underspending.