Aztec Micro-grid Power System

Grid Energy Storage and Harmonic Distortion Demonstration

Project Proposal

Submitted to: John Kennedy
Design Co. Ltd, San Diego, CA

Hardware:

Ammar Ameen
Bashar Ameen
Aundya Azarbarzin
Jeffrey Barrio
Nicholas Dobbs
Luke Vo
Sabri Germukly

Software:

Joshua Boss
Patrick Gutierrez
Abdullah Maarafi
Benjamin Santacruz
Daniel Valencia

Sponsor:
San Diego Gas & Electric

Table of Contents:

1. Introduction……3
   1.1. Abstract
   1.2. Project Description
2. Design……3
   2.1. Block Diagram
   2.2. Mockup Illustrations
   2.3. Performance Requirement
3. Testing and Verification……9
   3.1. Testing Procedures
   3.2. Benchmarks
4. Project Management……12
   4.1. Project Plan & Milestones
5. Budget……13
   5.1. Cost Analysis
6. Promotional Flyer……14

1. Introduction

1.1| Abstract:

SDG&E has requested for a working model of a miniature electrical network to demonstrate the effects and damage that nonlinear loads can cause to a power grid in comparison with traditional linear loads. This model will allow the user to provide any input from a linear or nonlinear load to see its harmonic contribution to the power grid through the Graphical User Interface. The GUI will output data collected from sensors which will be sent to a microcontroller and then to a Raspberry Pi. The Raspberry Pi will communicate wirelessly through a local area network, connecting to the PC that the GUI will display from.

1.2| Project Description:

Today's electrical power grid faces emerging challenges created by the effects of modern electronics and renewable energy. Nonlinear loads like AC motors, lighting ballasts, and switching power supplies draw non-sinusoidal currents, introducing unwanted harmonics into power systems. Additionally, renewable sources, like solar energy, produce and store DC power, requiring inverters to produce AC power for consumption. The sinusoids produced from these inverters often are below satisfactory and introduce additional harmonics into the system. As the total harmonic distortion increases, the total consumption of powered devices increases, creating unwanted heat and shortening electronics' lifespans. Awareness of this issue and research into mitigation are crucial to future performance.

San Diego Gas & Electric has asked San Diego State University to develop a working model of a miniature electrical network. This model will compare linear and nonlinear loads along with inverters varying in quality. Through the GUI, the user can monitor the harmonics from each load respectively, as well as control remote elements on the model that can mitigate the effects of nonlinear loads. The user can use the GUI to switch relays to choose between two inverters and three different loads, as well as an option for a line filter. The GUI associated with this model will allow the user to provide any input from a linear or nonlinear load in order to see its harmonic contribution to the power grid, along with monitoring the current and voltage of the system in the time or frequency domain. The data utilized by the GUI will be collected by various sensors within the model. These sensors will send the information to a microcontroller that will transmit to a Raspberry Pi. The Raspberry Pi microcomputer communicates wirelessly through a LAN to the PC containing the GUI.

The Aztec Micro-grid Power System team, also known as A.M.P.S., will be divided into two groups, Hardware and Software, to complete this effort. The Hardware team will develop the physical model that will serve as a mockup of a miniature electrical network. The Software team will design a graphical user interface that will serve as the model's monitoring and control system.

2. Design

2.1| Block Diagram:

2.1.1| Hardware:

The block diagram in Figure 1 below shows the electrical network and interconnections of the system. 120VAC provided from a mains outlet will charge a 12.8V/22Ah battery through the

model's compatible charger. The battery will power the simulated grid, excluding the Raspberry Pi and secondary microcontroller, which are powered by an external DC power supply. There are two inverters that can be selected one at a time to run off the battery. The input to these inverters is controlled by a relay on the front end. This relay will connect the battery for either charging or powering the grid. A relay on the output of the inverters will control which one supplies the load. One inverter will output a pure sine wave and the other a modified-sine wave, introducing different harmonics. The output of the selected inverter will run into three separate loads. The line to each load will consist of a relay to turn the load on and off, voltage and current sensors inline, and a relay selectable filter. Proposed loads are an AC motor driven fan, incandescent light, and LED light.

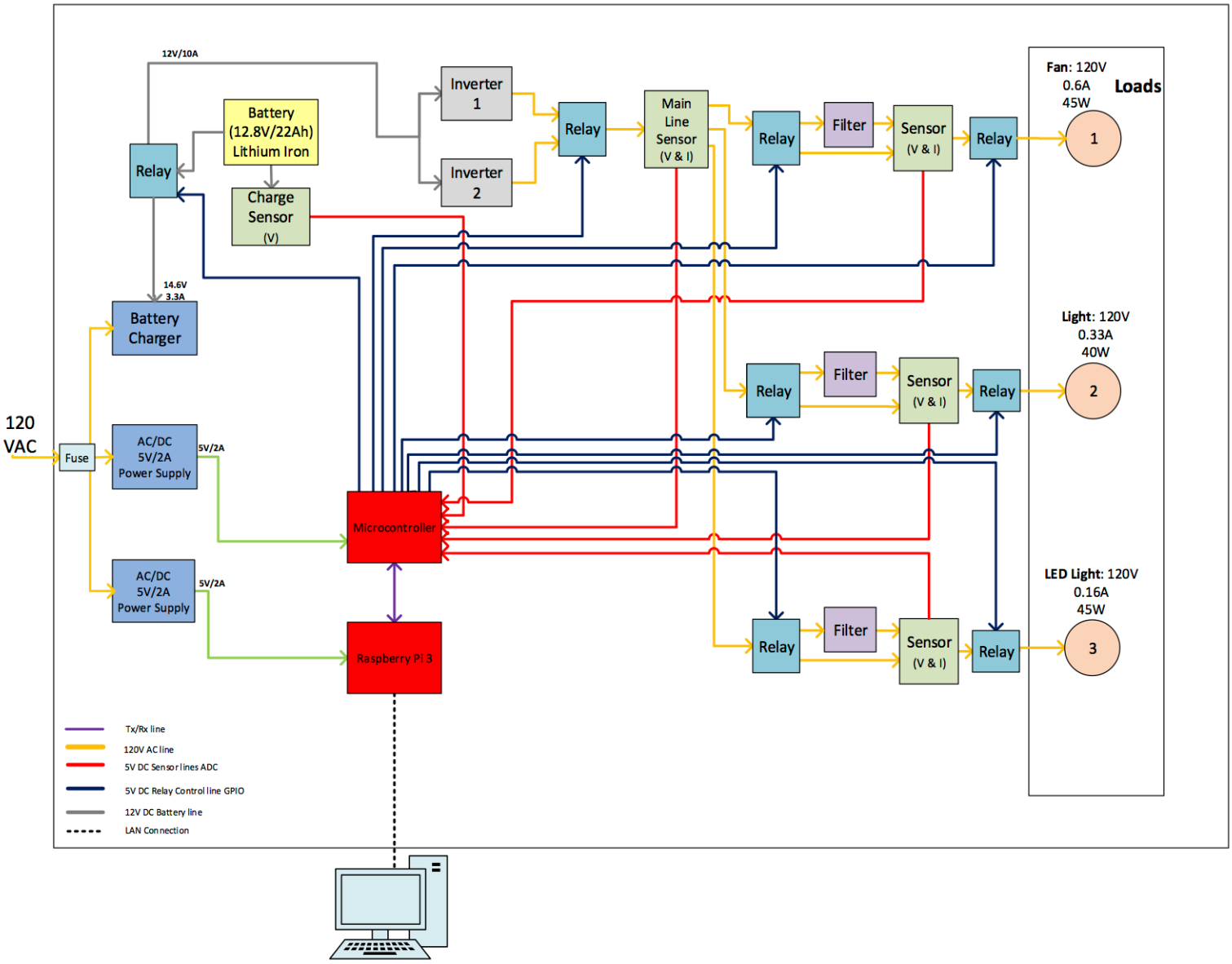


Figure 1: Hardware System Block Diagram

2.1.2| Software:

The following block diagram, Figure 2 shown below, demonstrates the physical interconnects of software components.

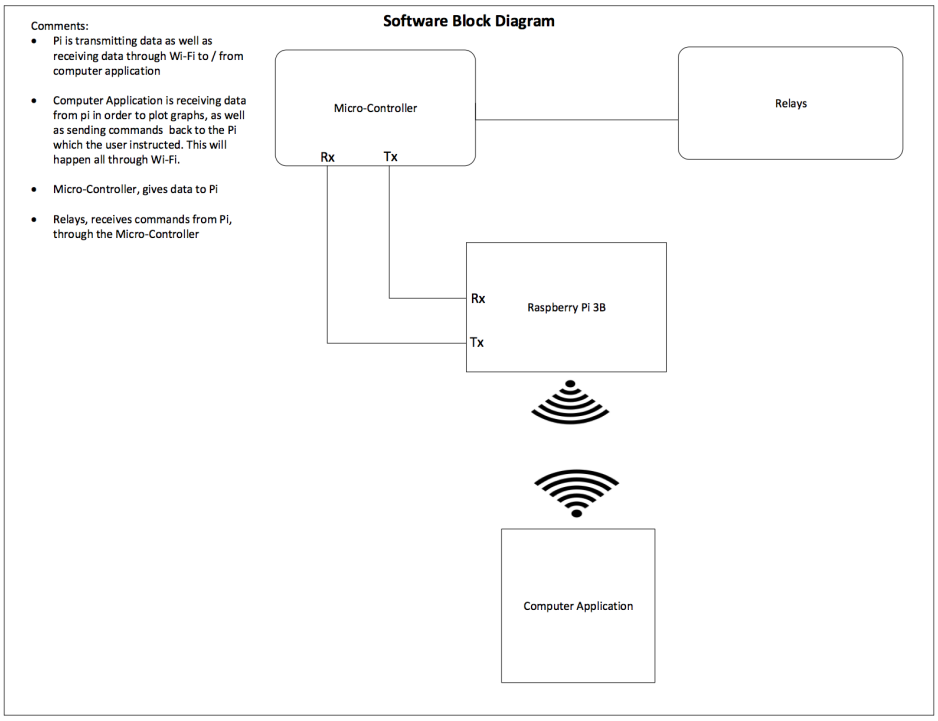


Figure 2: Software System Block Diagram

The software flow chart, Figure 3 shown below, demonstrates the logic of each function between the Raspberry Pi and PC.

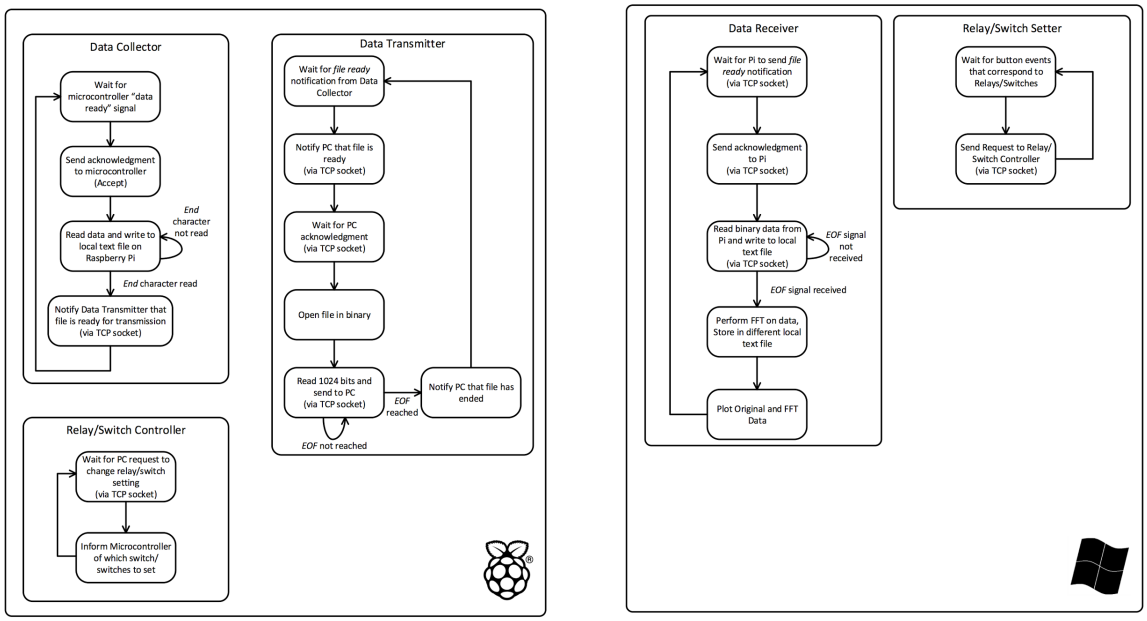


Figure 3: Logic Flow Chart for Functions within Raspberry Pi and PC

2.2| Mockup Illustrations:

2.2.1| Hardware Model:

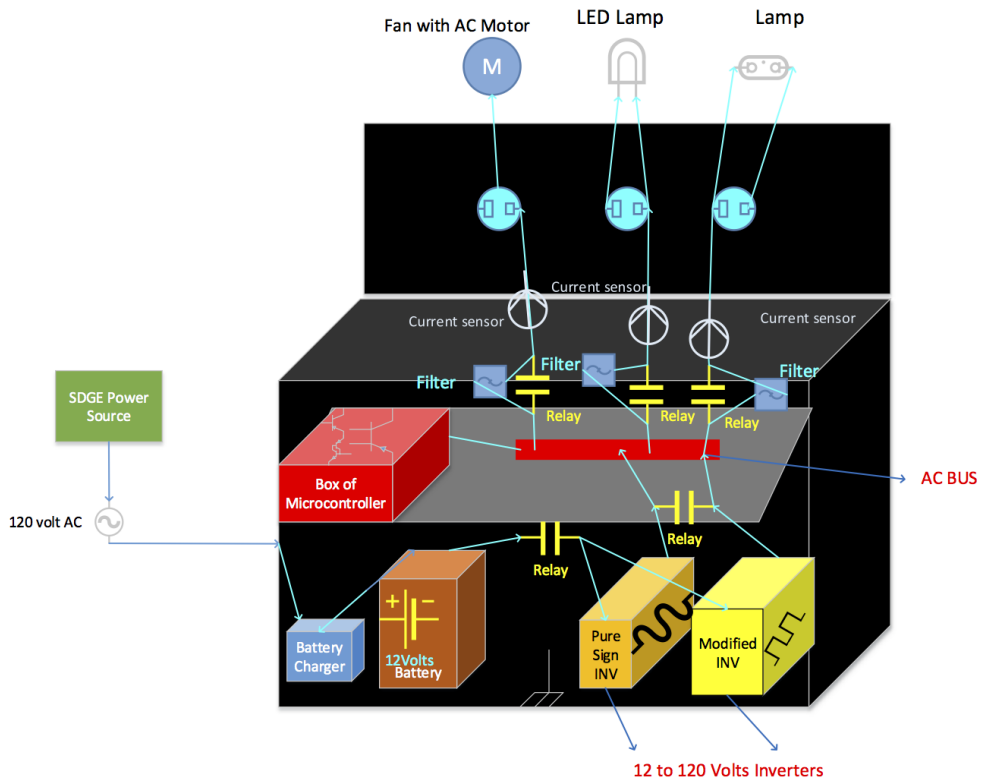


Figure 4: *Model of Carrying Case and Contents*

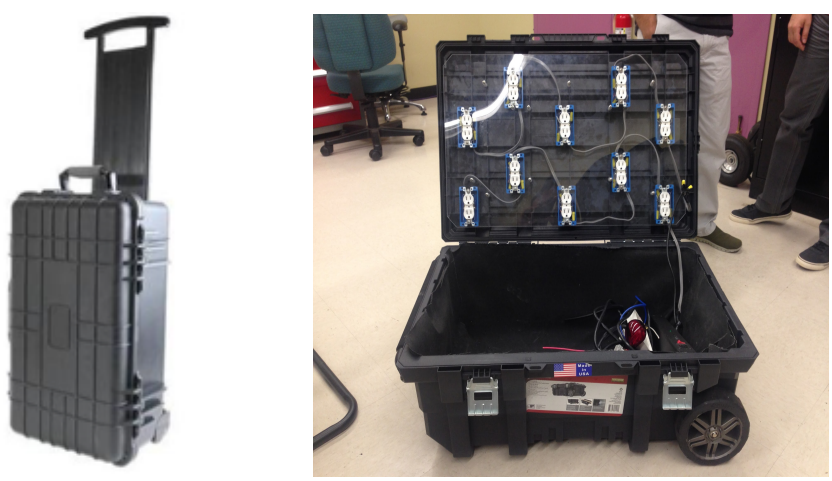


Figure 5: *Examples of Carrying Case*

2.2.2| Software GUI:

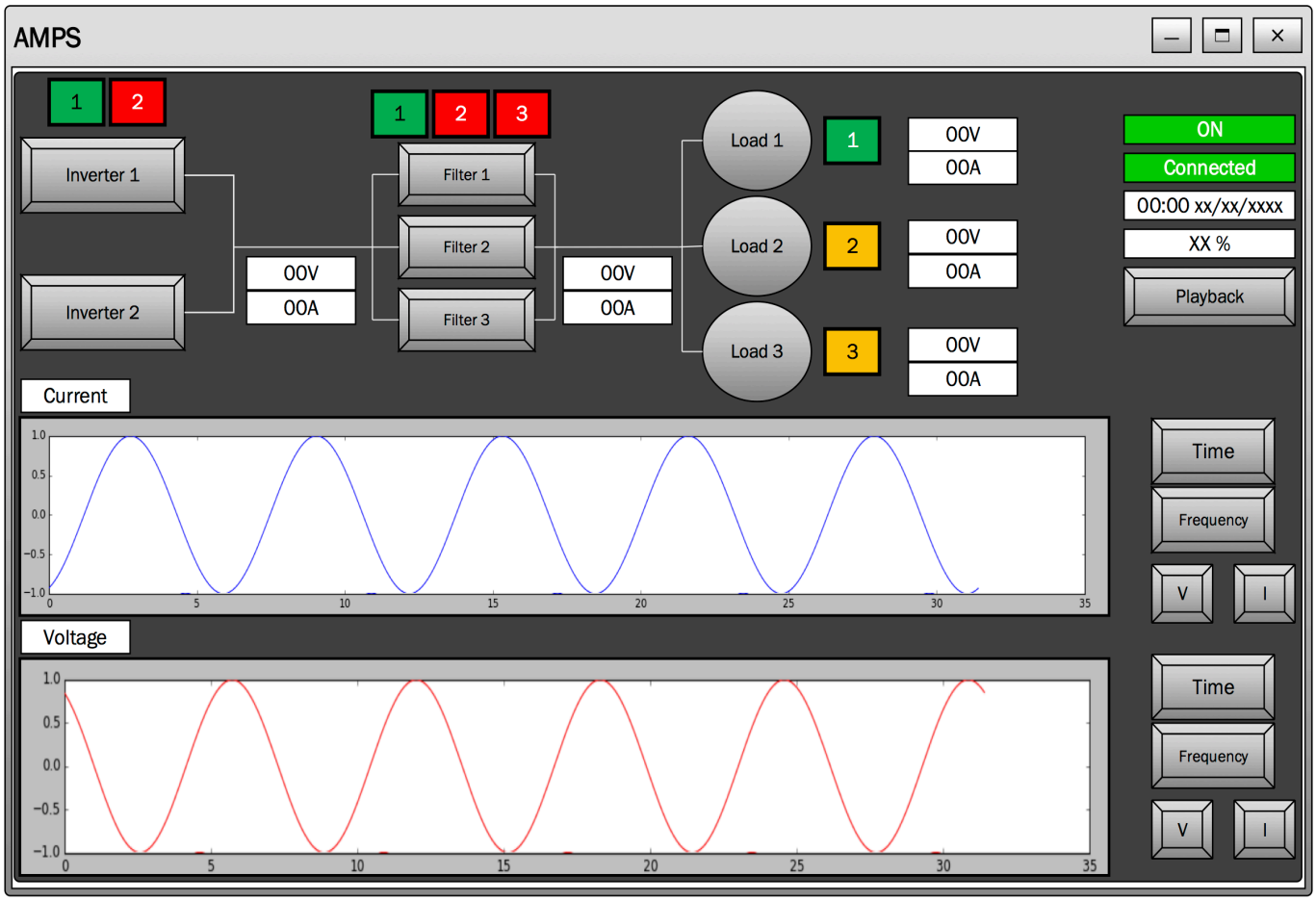


Figure 6: Graphical User Interface

2.3| Performance Requirement:

2.3.1| Hardware:

Battery operation	The system will operate for at least 1 hour with no connection to a power source. The internal battery will power the isolated AC circuit, instrumentation, and communication elements.
Sample Rate	The current and voltage sensors will operate at a sampling rate of 3 kHz. This will allow detection of signals up to the 15th harmonic.
Sensor Resolution	Current sensors will have a resolution of 2 mA at the output of the microcontroller. Voltage sensors will have a resolution of 1 V at the output of the microcontroller.

2.3.2 Software:

Data Processing/ Forwarding	The Raspberry Pi 3 will be used as the central data acquisition and data server unit. It will communicate with the microcontroller to receive sensor data. The Raspberry Pi will serve as a Wi-Fi access point to facilitate data transfer between a client machine by way of TCP sockets. Sensor data will be sent to the client machine at a rate of 2 Mb/s.
Refresh Rate	The GUI will display current and voltage waveforms at a minimum rate 0.5 Hz. This will be a sufficient amount of time to receive data from the Raspberry Pi and apply a Fast Fourier Transform (FFT) to display the harmonics. Performing a FFT on the client machine will allow the Pi to avoid sending additional data, which will decrease processing time.
Flow Control	The GUI allows the user to specify which of the inverters is in use, and select between the filtered and unfiltered power lines.
Historical Data and User Configuration	The GUI allows the user to view previous data from different loads and easily compares loads and/or filtered and unfiltered waveforms. It will also allow a user to quickly set all switches/relays to a previously configured state by loading a saved file.

3. *Testing and Verification*

3.1| Testing Procedures:

3.1.1| Hardware:

Verifying the harmonic content displayed is accurate	Obtain current and voltage waveforms from each load using lab equipment.	Voltage waveforms will be measured and recorded using an oscilloscope with standard probe.
		Current will be run through a shunt resistor equivalent to that used in the system. A differential oscilloscope probe will be used to measure the voltage across the resistor and record it.
		The data from these waveforms will be saved and exported from the scope as a .csv file.
		These files will be read into MATLAB and the resulting waveforms displayed.
		FFT will be performed in MATLAB and harmonic content observed.
	Obtain current and voltage waveforms from each load using system sensors.	Data for voltage and current waveforms will be recorded from the sensor into the microcontroller.
		Data from the microcontroller will be sent to the Raspberry Pi and to the connected PC.
		This data will be read in the designed system software and the waveforms displayed.
		This software will perform an FFT on these waveforms and the harmonic content obtained will be compared to that obtained from MATLAB. Accuracy of harmonic magnitudes must be within 2%.
Sensor Resolution Testing	Voltage accuracy testing	A known voltage will be applied across the voltage divider, with measured resistors, to the input of the sensor. Multimeters will be used to measure the applied voltage and the output of the sensor.
		The voltage will be changed in 1mV increments. Applied voltage and sensor output will be recorded.
		Measured voltage will be calculated to be used in software. Accuracy to the expected voltage will be compared. The resolution must be within 1mV.
	Current accuracy testing	A known current, from a power supply in constant current (CC) mode, will be applied through the sensor. Multimeters will measure current and output of the sensor.

		The current will be changed in 2 mA increments. Applied current and sensor output will be recorded.
		Measured current will be calculated in software. Accuracy to the expected current will be compared. The resolution must be within 2 mA.

3.1.2| Software:

Raspberry Pi Communication	Setup Raspberry Pi as Wireless Access point.	Establish an Ad – Hoc Connection for Raspberry Pi.
		Will join Wireless Network via Laptop Network Connections.
		Will verify that connection has been made by confirmation via PC Networks.
	The Raspberry Pi sends data via Wi-Fi to PC.	Run script on Raspberry Pi to send data (.txt/ASCII).
		Verify correct packet has been received.
		Open packet and verify data is correct.
	The Raspberry Pi sends and receives data in American Standard Code for Information Interchange (ASCII) to the microcontroller (MCU) via UART.	Send ASCII data from MCU to Raspberry Pi.
		Verify the correct data packet was received.
		Send data from MCU to Raspberry Pi.
		Verify correct data was received.
MCU Communication and Controls	The MCU will sample analog voltage and current signals via analog to digital conversion (ADC).	Check ADC is working correctly using a potentiometer.
		MCU samples voltages into the ADC at 3KHz.
		Check the sample is correct by plotting the data in the GUI.
	The MCU will control relays through GPIO pins.	Set pins to trigger relays.
		Verify that relays have triggered.
	The MCU will store sensor data into its memory temporarily, then immediately send the data to the Raspberry Pi when prompted to.	MCU will store 3.2 kB of sample data in memory.
		Send sample data to Raspberry Pi via UART with a baud rate of 115200.
		Verify correct sample and file size has been received.

PC/GUI	The GUI will be able to run on any Windows PC and behave as a standalone application.	Start GUI Application.
		Verify GUI starts correctly and has no errors on startup.
	The GUI will control the micro-grid power system via buttons, which will send commands to the Raspberry Pi.	Start GUI application.
		Click button on GUI for corresponding control (change relay, change load).
		Verify correct action has occurred, check relay status or load
	The GUI will indicate which load, filter, and inverter is selected.	Click button on GUI for load or filter selection.
		Verify indicator on GUI has changed states.
	The GUI will receive the data from the model and plot the voltage/current and the Fourier Transform of that signal.	Verify data is received.
		Plot sample data received from the Raspberry Pi.
		Display visual plot, and toggle between Voltage/Current and Fourier Transform of Signal.

3.2| Benchmarks:

3.2.1| Hardware:

1. The system will operate for at least 1 hour with no connection to a power source.
2. The current and voltage sensors will be sampled at 3 kHz.
3. Current sensors will have a resolution of 2 mA at the output of the microcontroller.
4. Voltage sensors will have a resolution of 1 V at the output of the microcontroller.

3.2.2| Software:

1. The MCU will sample data from any load and send to the the Raspberry Pi via UART at a data transfer rate of 14 kB/s.
2. Sensor data will be sent to the client machine at a minimum rate of 2 Mb/s.
3. Switch relays and filters between two states, on and off, via GUI.

4. Project Management

4.1| Project Plan & Milestones:

ID		Task Mode	Task Name	Duration	Start	Finish
1						
2			**Software Configuration**	**50 days**	**Mon 10/3/16**	**Fri 12/9/16**
3			**Raspberry Pi (Rpi) Communications and Controls**	**16 days**	**Mon 10/3/16**	**Mon 10/24/16**
4			Set up Rpi as wireless access point	2 days	Mon 10/3/16	Tue 10/4/16
5			Configure Rpi to send data via wireless comm to PC	6 days	Wed 10/5/16	Wed 10/12/16
6			Configure Rpi to communicate with MCU (UART)	12 days	Fri 10/7/16	Mon 10/24/16
7			**MicroController Unit (MCU)**	**22 days**	**Mon 10/3/16**	**Tue 11/1/16**
8			Configure MCU sampling and storing	13 days	Mon 10/3/16	Wed 10/19/16
9			Configure MCU to communicate with Rpi (UART)	7 days	Wed 10/19/16	Thu 10/27/16
10			Configure MCU relay switch control	7 days	Mon 10/24/16	Tue 11/1/16
11			**PC/GUI Development**	**50 days**	**Mon 10/3/16**	**Fri 12/9/16**
12			Configure GUI to select components via button/send command	10 days	Mon 10/10/16	Fri 10/21/16
13			Plot sampled oscilloscope data values on GUI	5 days	Mon 10/24/16	Fri 10/28/16
14			Configure playback feature for data values on GUI	6 days	Fri 10/28/16	Fri 11/4/16
15			Configure PC to control relay switches	7 days	Tue 11/1/16	Wed 11/9/16
16			GUI development (Python)	50 days	Mon 10/3/16	Fri 12/9/16
17			**Hardware Configuration**	**50 days**	**Mon 10/3/16**	**Fri 12/9/16**
18			Initial investigation of harmonic content and waveforms	2 days	Mon 10/3/16	Tue 10/4/16
19			Simulate waveforms on programs	2 days	Mon 10/3/16	Tue 10/4/16
20			Research components: sensors, relays, microcontrollers	11 days	Mon 10/3/16	Mon 10/17/16
21			Sensor prototyping	8 days	Mon 10/17/16	Wed 10/26/16
22			Sensor qualification testing	6 days	Thu 10/27/16	Thu 11/3/16
23			Sensor and MCU integration testing	5 days	Fri 11/4/16	Thu 11/10/16
24			Build and test power grid model without relay function	15 days	Tue 10/11/16	Mon 10/31/16
25			Integration of relays to power grid model	5 days	Tue 11/1/16	Mon 11/7/16
26			Sensor to power grid system integration	3 days	Tue 11/8/16	Thu 11/10/16
27			**PCB**	**22 days**	**Thu 11/10/16**	**Fri 12/9/16**
28			Design rev0 PCBs	7 days	Thu 11/10/16	Fri 11/18/16
29			Test rev0 PCBs	3 days	Mon 11/21/16	Wed 11/23/16
30			Design rev1 custom PCBs	4 days	Thu 11/24/16	Tue 11/29/16
31			Test rev1 custom PCBs	3 days	Wed 11/30/16	Fri 12/2/16
32			Design rev2 custom PCBs	3 days	Mon 12/5/16	Wed 12/7/16
33			Test rev2 custom PCBs	2 days	Thu 12/8/16	Fri 12/9/16
34			**Integration and Testing Event**	**16 days**	**Fri 11/18/16**	**Fri 12/9/16**
35			Build Unit Housing Box	10 days	Fri 11/18/16	Thu 12/1/16
36			Integrate hardware and software components	10 days	Fri 11/18/16	Thu 12/1/16
37			Finalizing project	6 days	Fri 12/2/16	Fri 12/9/16
38						
39						
40			Sample data on MCU ADC at 3KHz and transmit to Rpi	0 days	Thu 10/27/16	Thu 10/27/16
41			Power system end-to-end from battery to load without relays	0 days	Mon 10/31/16	Mon 10/31/16
42			Press button on PC to receive data from Rpi and plot	0 days	Tue 11/1/16	Tue 11/1/16
43			Press button on PC to tell Rpi to tell MCU to control relay switches	0 days	Mon 11/14/16	Mon 11/14/16
44			Successfully demonstrate PC control of components selection in completed system	0 days	Mon 11/21/16	Mon 11/21/16
45			Successfully integrate functional whole system into final model	0 days	Thu 12/1/16	Thu 12/1/16
46			Successfully demonstrate full peripheral capabilities on custom PCB when integrated with power grid model	0 days	Fri 12/9/16	Fri 12/9/16
47			Successfully demonstrate end-to-end system testing of voltages and currents to GUI	0 days	Fri 12/9/16	Fri 12/9/16
48			Final website deployed	0 days	Mon 12/12/16	Mon 12/12/16
49			Senior Design Presentation	0 days	Tue 12/13/16	Tue 12/13/16

5. Budget

5.1| Cost Analysis:

The allocated budget from SDG&E is $1000. The following list below is our projected spending. Most of the budget will be employed in the hardware side which takes up 40% of the budget. The enclosure will encompass the carrying case for the model along with the outlets and the fans used for internal temperature control. The loads will be an AC motor driven fan, incandescent light, and an LED light. The software side incurred smaller expenses due to the main cost being micro-controllers, taking up 18% of the budget. Furthermore, our leftover balance of $200 will be later utilized for miscellaneous components including replacement parts.

Sensor Components	$100
Inverters	$130
Relays	$100
Battery	$0
Microcontrollers	$180
Loads	$40
Enclosure	$250
Miscellaneous Components	$200
Total Requested:	**$1000**

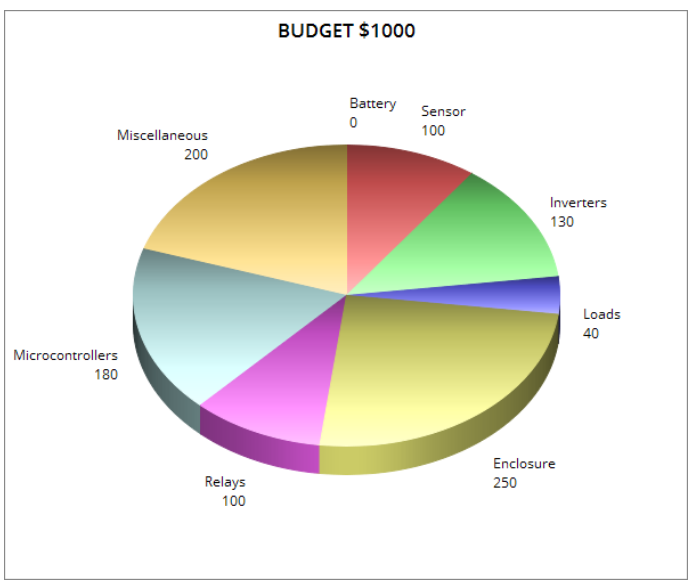
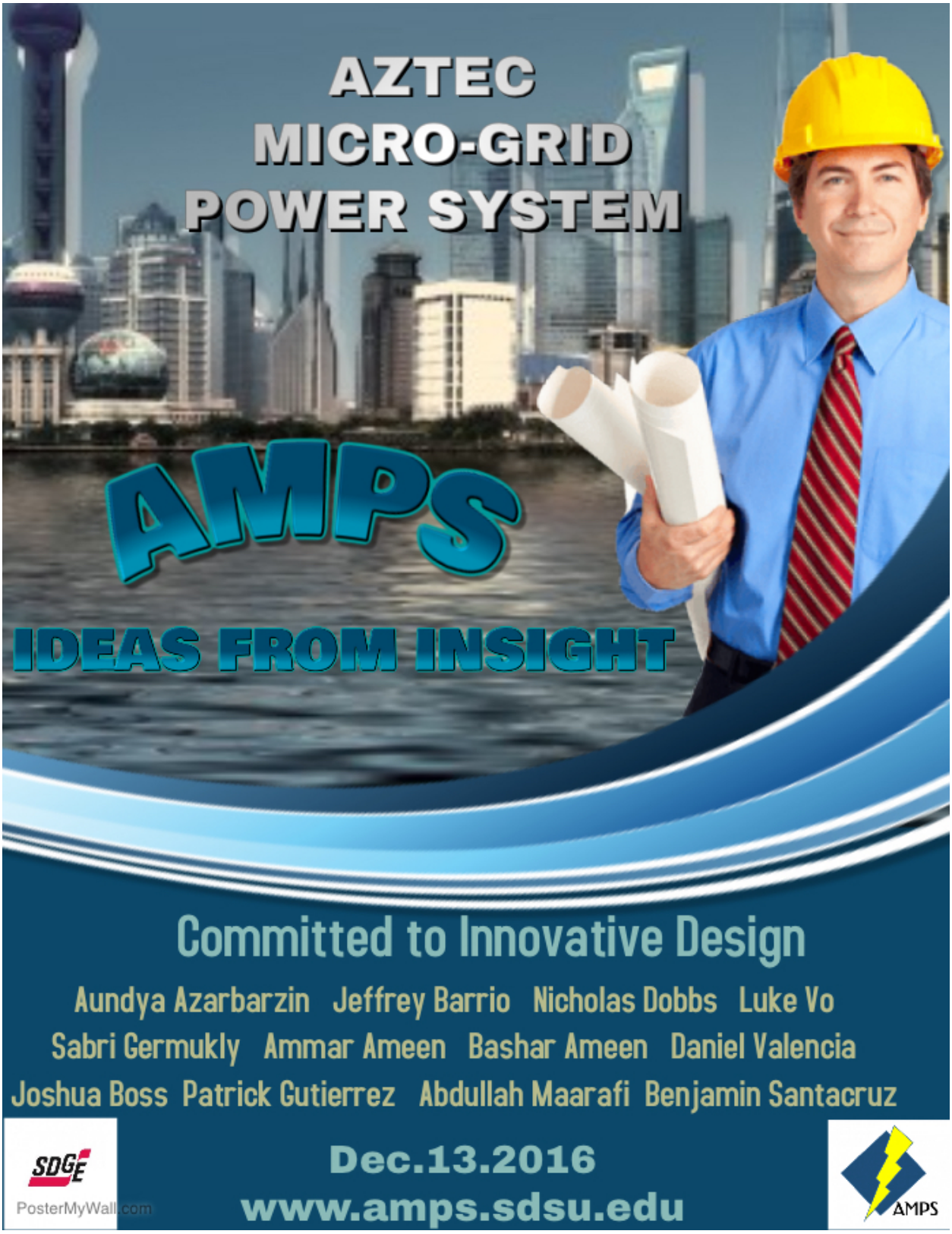


Figure 7: Pie Chart Representing $1000 Allocated Budget

6. *Promotional Flyer*