intelliGrid

Bryan Rodriguez
Derek Flesher
Edmond Basilan
Emmanuel Demonteverde
Gabriel Duran
Harrison Snodgrass
Joshua Pritts
Kysa Tran
Lawrence Meyer
Lucas Erpelding
Mehmet Isik
Rachel Shrum
Tony Nguyen

SDGE
A Sempra Energy utility®

San Diego State University

Abstract and Description

Abstract:

San Diego Gas & Electric, SDG&E, is introducing the initial implementations of the Smart Grid design for their electrical system. To effectively communicate with customers about their Smart Grid efforts, the IntelliGrid team is creating a model that will integrate software and hardware components to demonstrate a realistic electrical network that will represent SDG&E's vision. The model will simulate the ability to generate and dynamically dispatch distributed energy resources to support system under both steady state and emergency conditions. It will also display the implementation of a self-healing network concept and show the effects of renewable energy on the grid.

Project Description:

Hardware

In order to demonstrate SDG&E's Smart Grid efforts, a model of San Diego will be built demonstrating the concept of a self-healing network. The model will have a sub-station, a high density and commercial area, as well as residential areas. As seen in Figure 1, the substation will feed energy into the grid which the commercial and residential areas will consume. Apart from the conventional energy produced, wind turbines will provide energy, and any excess energy will be stored in a battery.

In order for the model to accurately represent SDG&E's energy grid, fault isolating devices controlled by the utility and unpredictable loads will be a part of the model. Switches, reclosers, and transformers will react when a fault is present. Unpredictable loads include roof-top solar panels and charging of electric vehicles. Both of these cause changes to the flow of the grid, possibly creating faults. Other faults that will be accessible in the model are Mylar balloons getting caught in overhead lines and a backhoe hitting an underground line.

The model will be built using several microcontrollers each controlling a small portion of the overall model, as seen in Figure 2. Communication between the tablet and the hardware model will be established through Wi-Fi. The Wi-Fi module on the box will use SPI to talk to the master microcontroller. It, in turn, will communicate with its slave microcontrollers through I2C.

The energy flow will be illustrated through LEDs. Branches will lead to the commercial and residential loads in the city model. Different colors will represent normal flow, green energy flow, and faults. Lastly, an LCD screen on the model will display the time of day, season, and wind speed simultaneously with the app.

A mock-up model and energy flow is shown in Figure 8. The model will represent a central area of San Diego, with the bay, PETCO Park, and the Air & Space Museum serving as distinct local landmarks. The landmarks, commercial buildings and residential houses will be 3D printed. Two other views of the model are shown in Figure 9.

The model will be placed on an 80/20 box made of aluminum. The dimensions will be 2 ft. x 3 ft. with a 5 in. height. Figure 10 shows the box, with the lid transparent only to show the back of the box where the hinges will be. The model will have the ability to be lifted, for users to view the electronics of the model.

Software

To control the grid, we are creating an application on the Android operating system. We will be connecting to the grid through a Wi-Fi connection where we will be able to control the overall model. The app will be controlled on a tablet that will have the simulation of the power flow running in the background. The app will also display a user friendly interface where the user can adjust the speed of time on the grid while also being able to control what season it takes place in. The user will also be able to create different fault scenarios while the app will display how the grid performs its self-healing process. The app will be able to display data and graphs of the fault analysis which will show real and apparent power through each transmission line. To display the app on an external screen, we decided to choose a tablet that can output HDMI.

Figure 3 displays a diagram of how the app and hardware will function as a whole. In Figure 4 we see how power is distributed from SDG&E to the different areas which includes the residential, commercial, and industrial. The diagram shows how the simulation would look like when it is in ideal conditions with no faults. Figure 5, on the other hand, shows how a fault in the grid disrupts the flow of power. We see how power flows through the industrial and commercial side into the residential areas to compensate. Figure 6 shows how power is distributed among the different areas throughout the course of the day. While Figure 7 shows the power generated by renewable energy and how the net power used by the grid is affected.

Block Diagrams

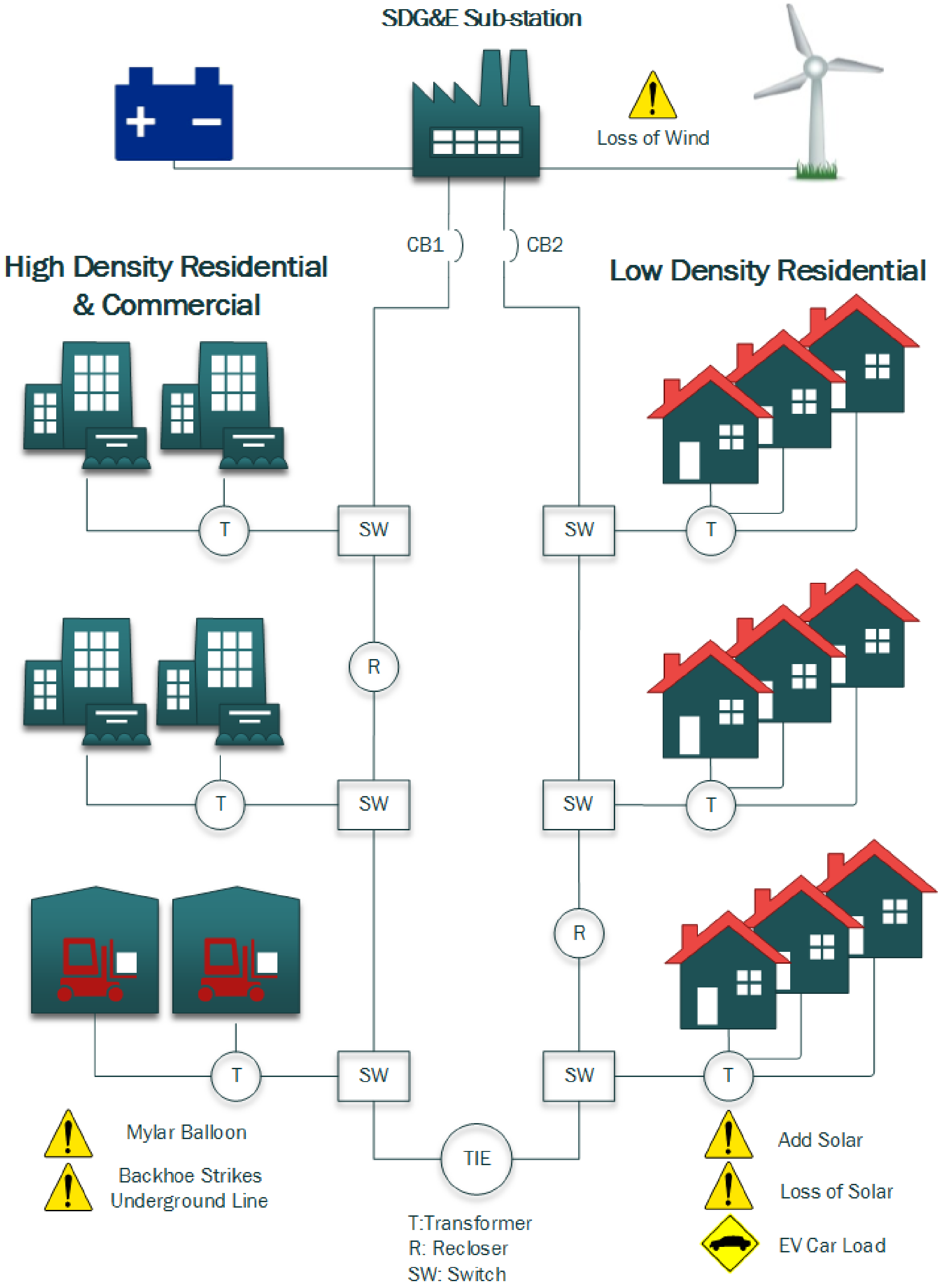
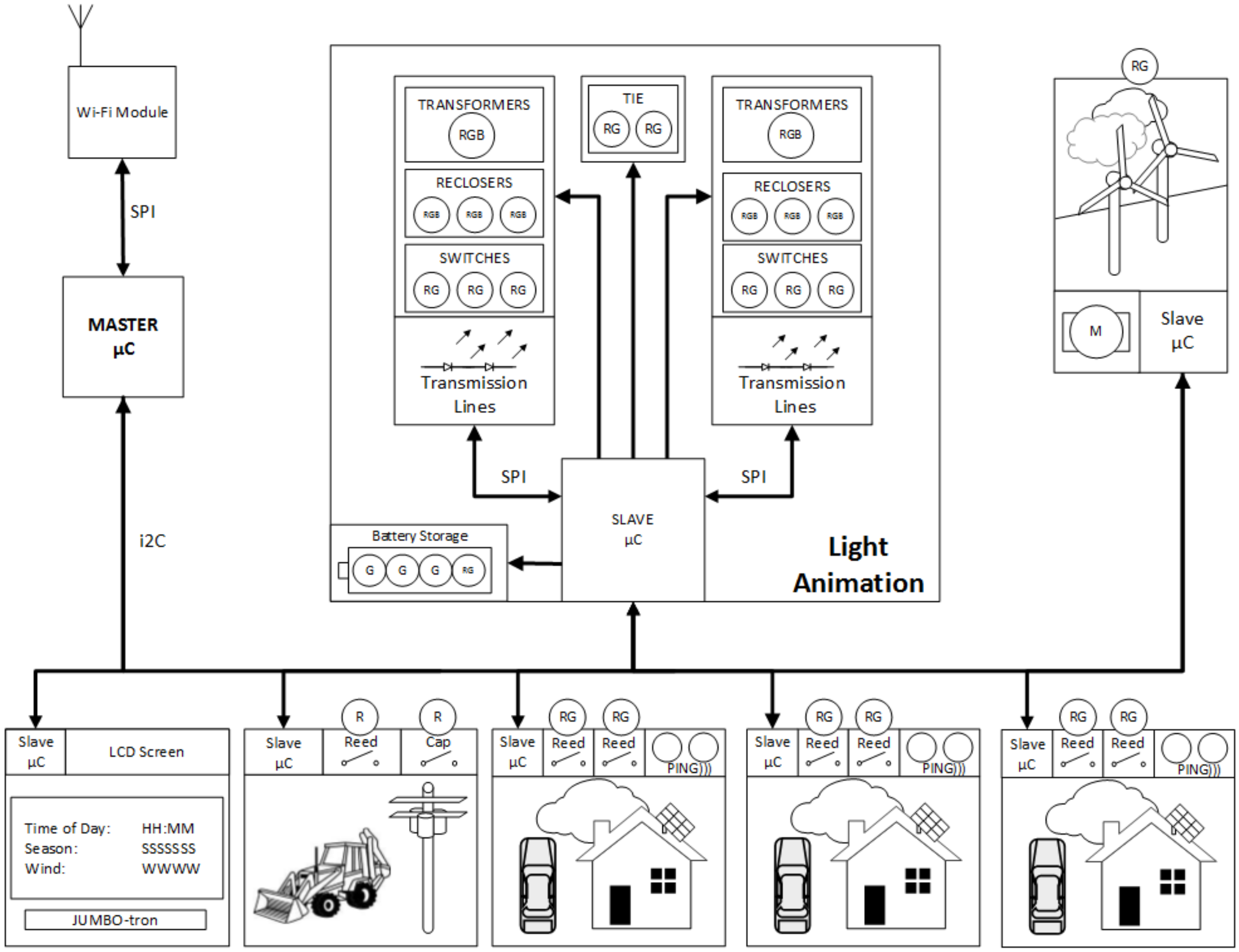


Figure 1 - Smart Grid Model



LEDs

RGB = Red Green Blue

RG = Red Green

R = Red

G = Green

Figure 2 - Model System Design

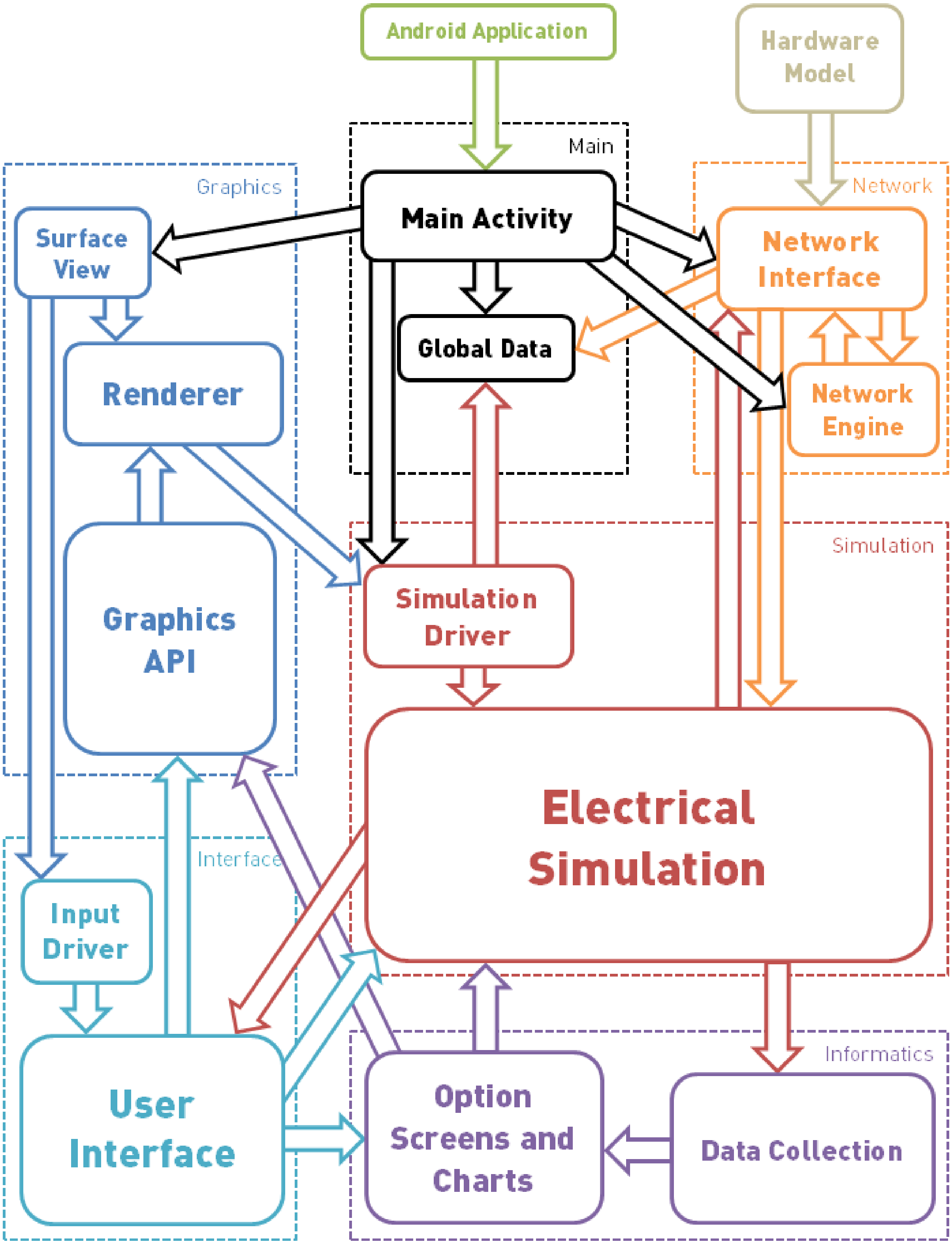


Figure 3 - Software Block Diagram

Supporting Illustrations

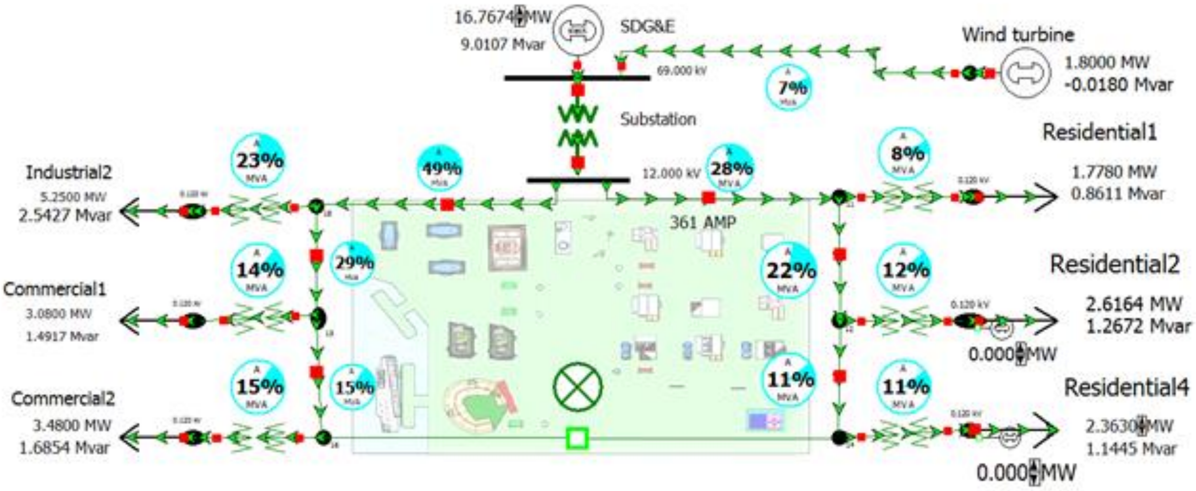


Figure 4 - The power flow simulation should look similar to this in ideal conditions

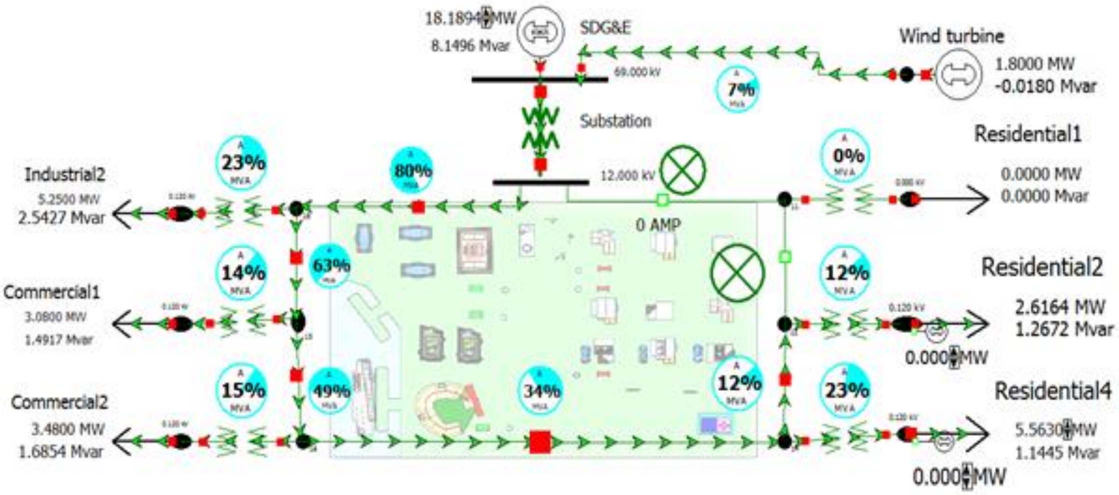


Figure 5 - If a fault is induced in the system, the power flow simulation will change as shown

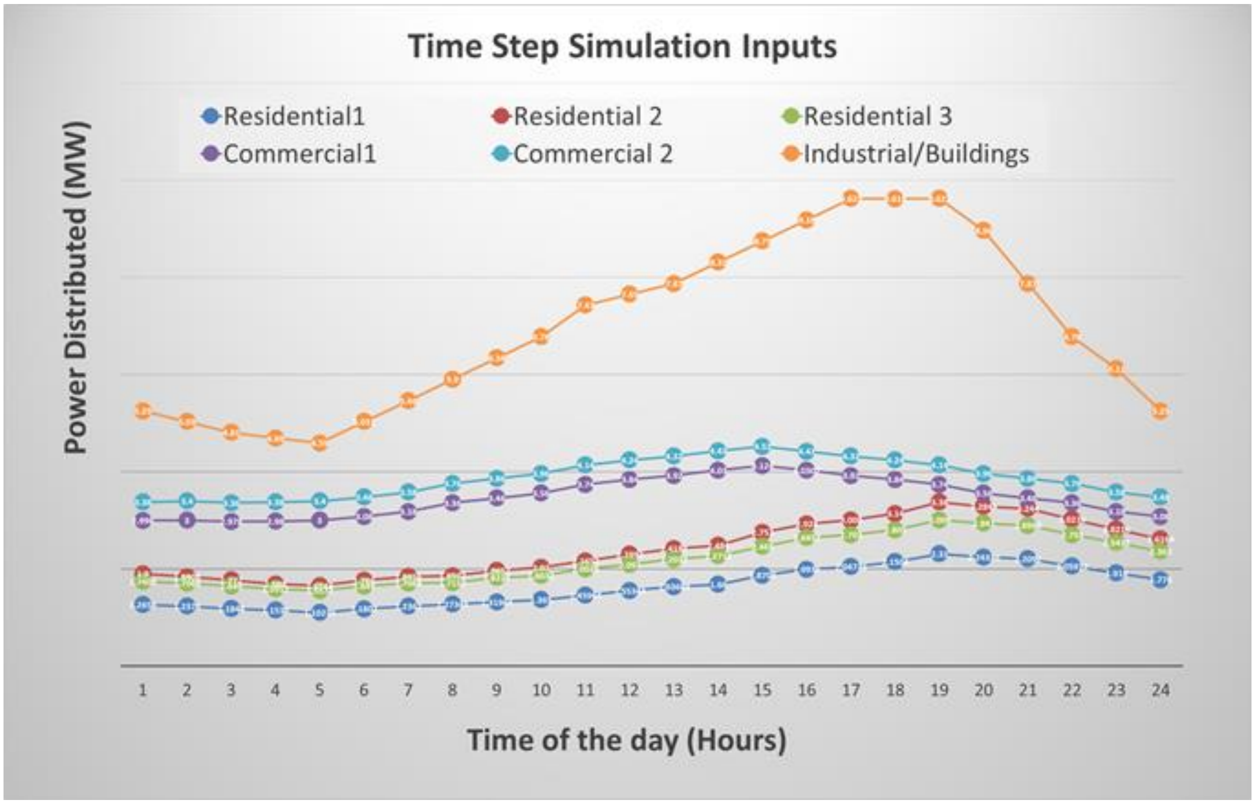


Figure 6 - On a 24 hour time step simulation, the power is distributed as shown

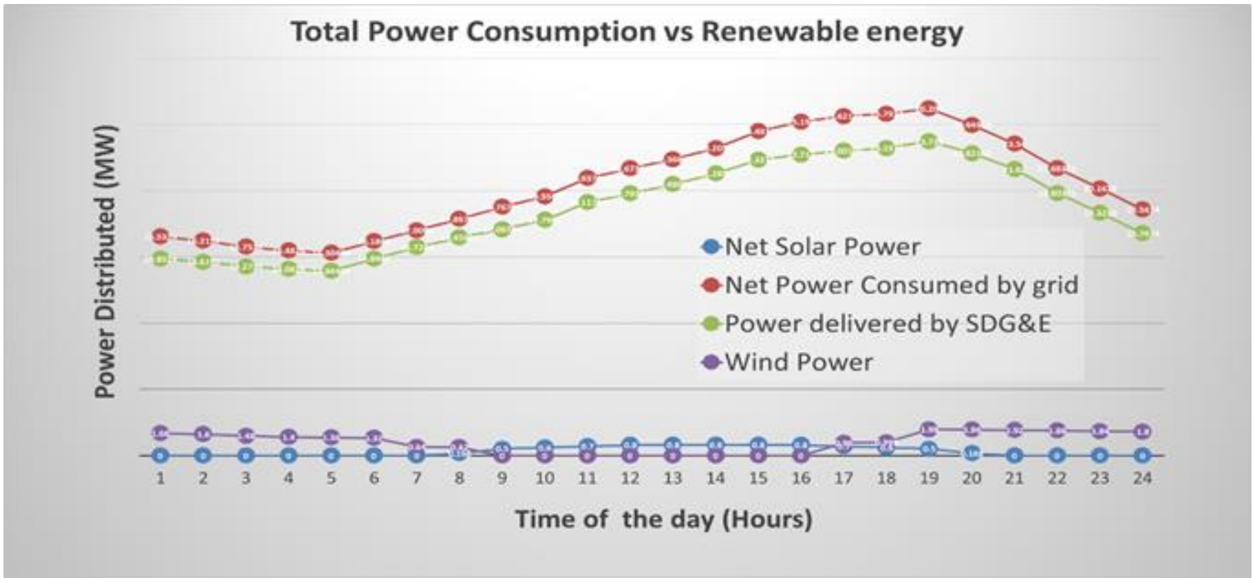


Figure 7 - This graph shows the power being generated by renewable energy, and how the net power consumed by the grid is being affected.

Mock-Up Illustration

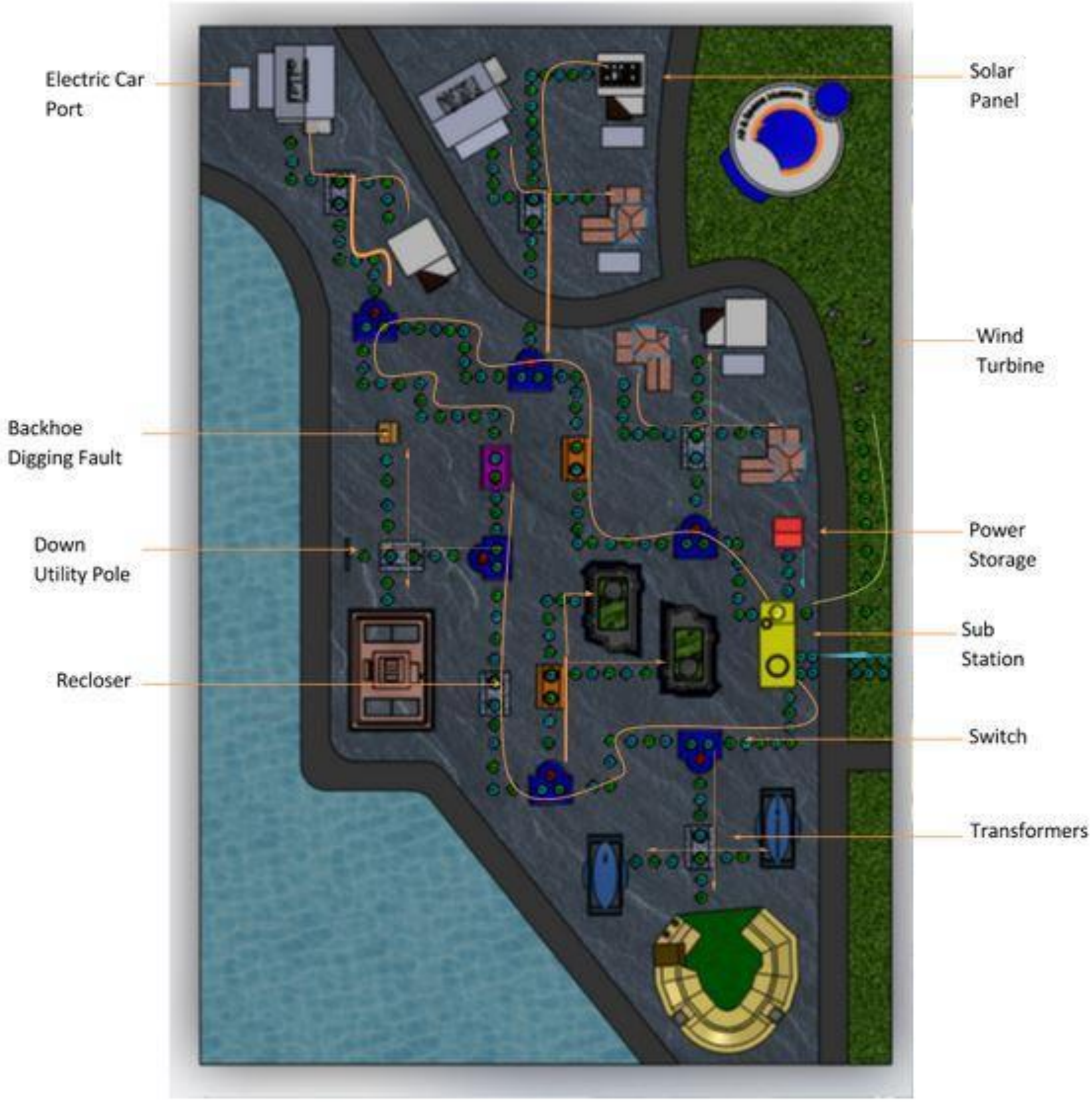


Figure 8 - Flow of Energy in the Model

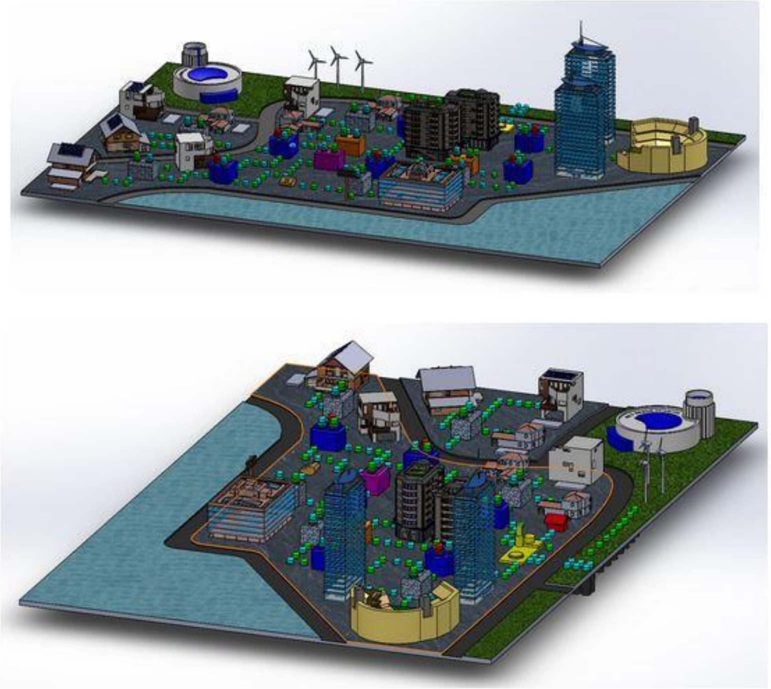


Figure 9 - Side Views of the Model

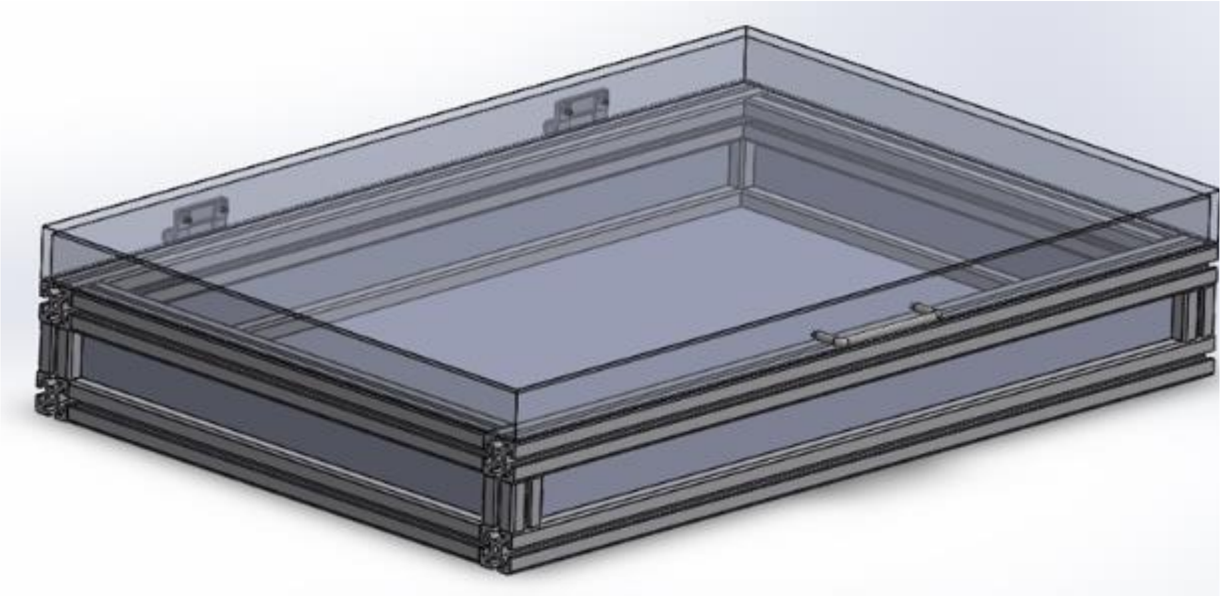


Figure 10 - 80/20 Box

Project Specifications

Smart Grid (Light Animation) Module

1. **Wi-Fi Module**
    - Operates in the voltage range between 2.8V – 3.6V
    - Establishes a communication embedded web server
    - Application throughput is 4500 kbps
    - Able to communicate with the tablet within 60ft

2. **Wind Turbine**
    - Spins at a rate between 40 rpm and 120 rpm as do real wind turbines
    - Speed is controlled by two push buttons on the model and also by the app
    - There are four speed settings; the speed level is displayed on the LCD screen located on the model

3. **Graphic LCD (GLCD) Screen**
    - The GLCD screen will display current season, wind speed, and an option of displaying the current time of the simulation
    - The GLCD data will be updated by the control tablet
    - Communications will be executed via the I2C bus
    - The GLCD will be controlled using a PIC18 microcontroller

4. **Light Animation**
    - The status and color of the LEDs show the proper flow of power in the grid
    - Green LEDs show the flow of green energy, the fully charged status of the battery… etc.
    - Blue LEDs show the flow of conventional energy in the grid
    - Red LEDs show faults and dead battery status
    - Yellow LEDs show maximum capacity of transformers and re-closers in the grid
    - Off status of LED shows the disconnected portion of the grid

5. **House Module**
    - Detects placement and removal of an electric car and of a solar panel
    - Detection will be visualized by an LED (red corresponds to electric vehicle, green corresponds to solar panel)
    - Detects a hand over the solar panel to initiate cloudy conditions
    - Proximity detection will only occur when solar panel is added to the system
    - Will have 2-way communication with its master controller

6. **80/20 Box**
    - Frame will be 2ftx3ft and 5 inches tall
    - Bamboo cover plates to be lightweight, durable, and environment friendly
    - 2 friction hinges attached to a 1 inch 80/20 T-slot lid on the top of the frame
    - Aluminum cover will be placed on the top of the model
    - Bottom side of the frame will serve as storage for cars, roofs…etc. and will also hold the power supply
    - Microcontrollers, LED lights, wires, and PCBs will be placed under the lid

Smart Grid Tablet Application

1. **Fully interactional with model**
    - Connected through local Wi-Fi
    - Responds to model's I/O devices
    - App commands control of LED flow on model
    - Simultaneous with LCD display on model
    - Controls overall time
    - Simulates each fault

2. **User Interface**
    - User friendly
    - Simple yet rich with content
    - Displays full demonstration of model in different levels
        - Basic power flow level
        - Illustrative display that is similar to physical model
    - Displays interactive power distribution graphs

3. **Tablet**
    - Program developed for Android 4.4 KitKat
    - 10 inch screen
    - Can be presented on projector via HDMI

Testing Procedures:

Hardware

1. **Wi-Fi Module**

   To test the ability of the Wi-Fi Module to establish communication with the embedded web server, the Wi-Fi G Demo board is used along with the PICkit 3 as the hardware and the MPLAB X IDE and MPLAB XC32 Compiler as the software. To test the throughput of the Wi-Fi module, we will install and use the TamoSoft Throughput Test software on a laptop. This utility continuously sends TCP and UDP data streams across the network and computes important metrics, such as uplink and downlink throughput values and packet loss.

2. **Wind Turbine**

   Testing of the wind turbines will begin with establishing control of the motor by the push buttons. The oscilloscope will be used to check the PWM signal applied to the motor in order to obtain the proper duty cycle for each speed level. Control of the wind turbines by the master microcontroller will be verified by a test program loaded on the master microcontroller.

3. **GLCD**

   Verification of operation will be performed by sending the GLCD microcontroller all update commands and checking the GLCD for proper updates.

4. **Light Animation**

   Once power is applied to the model, the Smart Grid (Light Animation) will demonstrate all of the LED configurations to verify functionality. If an LED does not illuminate, remove and replace LED.

5. **House Module**

   Testing of the house module will begin with verifying that the microcontroller reads the opening and closing of the reed switches. Two LEDs per switch controlled by the microcontroller will light up, one when the switch is open and the other when it is closed. After a cutoff distance is chosen for the proximity sensor, the distance will be tested by lighting an LED for a few seconds. However, no change will be seen when one of the reed switches is open.

Software

- The Software should be able to receive and transmit data to the Wi-Fi Module.
  - Constant communication (Controls all hardware).
- The design of the grid in the app should be almost identical to the physical grid.
- The Software should be able to change to three different base settings.
  - Seasons (Winter, Spring, Summer).
  - Speed (Normal, Fast, Super Fast, Pause).
- According to every base setting, the software should be able to analyze any fault or multiple faults at any moment.
  - Must be obvious when there is a fault.
  - Power Flow must be stopped at the specific location, and power must be redirected and initialized in steps (When a fault can be automatically fixed).
  - Must be able to show all the disturbance data and record it for future graphing.
- All the data used as input must be recorded in a sheet to be analyzed at any moment.
  - Graph all the data given.
  - Graphs should be available at any moment.
  - Data sheets of all the load demands.

Procedures and Benchmarks

Hardware Benchmarks

The different hardware systems that comprise the visual and interactive portion of the model must be completed individually before the larger system is constructed. Load distribution, solar and wind generation, and different types of faults will be simulated using RGB LEDs, read switches, sonic range finders, an LCD, DC motors, push buttons, and multiple microcontrollers. Once these systems have been tested individually the next benchmark is to integrate all the systems together under the control of a single microcontroller. Control over the system must then be established by the App via a Wi-Fi module mounted to the model. By this time the city scape will have been 3-D printed and the entire model can be constructed. After construction final testing will be conducted.

Network Hardware Link

1. Wi-Fi module implementation
2. TCP listen/receive/send loop
3. Accepts and sends signals according to agreed-upon protocol
4. Connected to at least 1 LED (input) and 1 switch (output)
5. Tested to work with a dummy signal
6. Finalized network protocol

Software Benchmarks

Program Outline

1. Android program shell
2. Program control loop
3. Class structure skeleton (empty classes and interfaces)
4. Global structures defined (important: network input/output buffers)
5. Javadoc started

OpenGL API

1. Sprite-based graphics engine customized for this project
2. Fixed function pipeline renderer
3. OpenGL ES 2.0
4. Tested to function properly
5. Tested with a torture test to determine rendering limits

Network Interface

1. Separate thread
2. TCP listen/receive/send loop
3. Asynchronous input/output buffers
4. Does not thread-lock, does not write to simulation
5. Tested to work with the hardware link

Electrical Simulation

1. Runs off of master program loop
2. All time-based calculations are based on a float amount input
3. Framerate-independent
4. Complete simulation; runs as a closed loop on its own
5. Does not need to be thread-safe
6. Reads network input buffer, writes to network output buffer (leave function stubs)
7. Modular nodes; individual wires or devices should be controllable and changeable
8. Has a single instance with the specific hardware layout we are using
9. Allows time-based (linear) interpolation
10. Allows any number of faults which can go on any node
11. All network properties/statistics should be run-time configurable
12. Programmable statistics profile to define scalars and constants
13. Default statistics profile should match SDG&E’s data
14. Necessary global data should be defined (e.g. time of day, temperature)
15. Tested to run to spec

Main User Interface

1. User input event handler
2. Implement planned interface functionality and graphics
3. Does not need to be thread-safe
4. Runs off of master program loop
5. All time-based graphics and animations are based on a float amount input
6. Should be instanceable
7. All main-screen functionality (faults, in-line data, display layers, electricity flow, etc.)
8. Uses our OpenGL API

Auxiliary User Interface

1. All options screens implemented
2. Graphs and infographics implemented
3. Graphs should be run-time configured (e.g. subtract X from Y, compare A and B)
4. Runs off of master program loop
5. May use any rendering techniques
6. Main screen is not rendered while options screens are up, but simulation keeps running

Software Finalization

1. Entire program works properly, up to spec, with no bugs
2. Runs with acceptable performance
3. Documentation written; in-app help if necessary
4. Deployed on our group tablet

Project Plan and Milestones

Hardware Plan

Each member of the hardware team is responsible for a different visual or interactive part of the model. The tasks required to make the model interactive are design and construction of the RGB LED distribution lines, battery storage, wind turbines, houses with solar panels and electric cars, the LCD screen that displays the status of the model, the fault caused by a dig, and the fault caused by a foil balloon. Group members are also tasked with integrating the different interactive elements together to communicate with the App via Wi-Fi, and with designing, then printing, the landscape and buildings using CAD software.

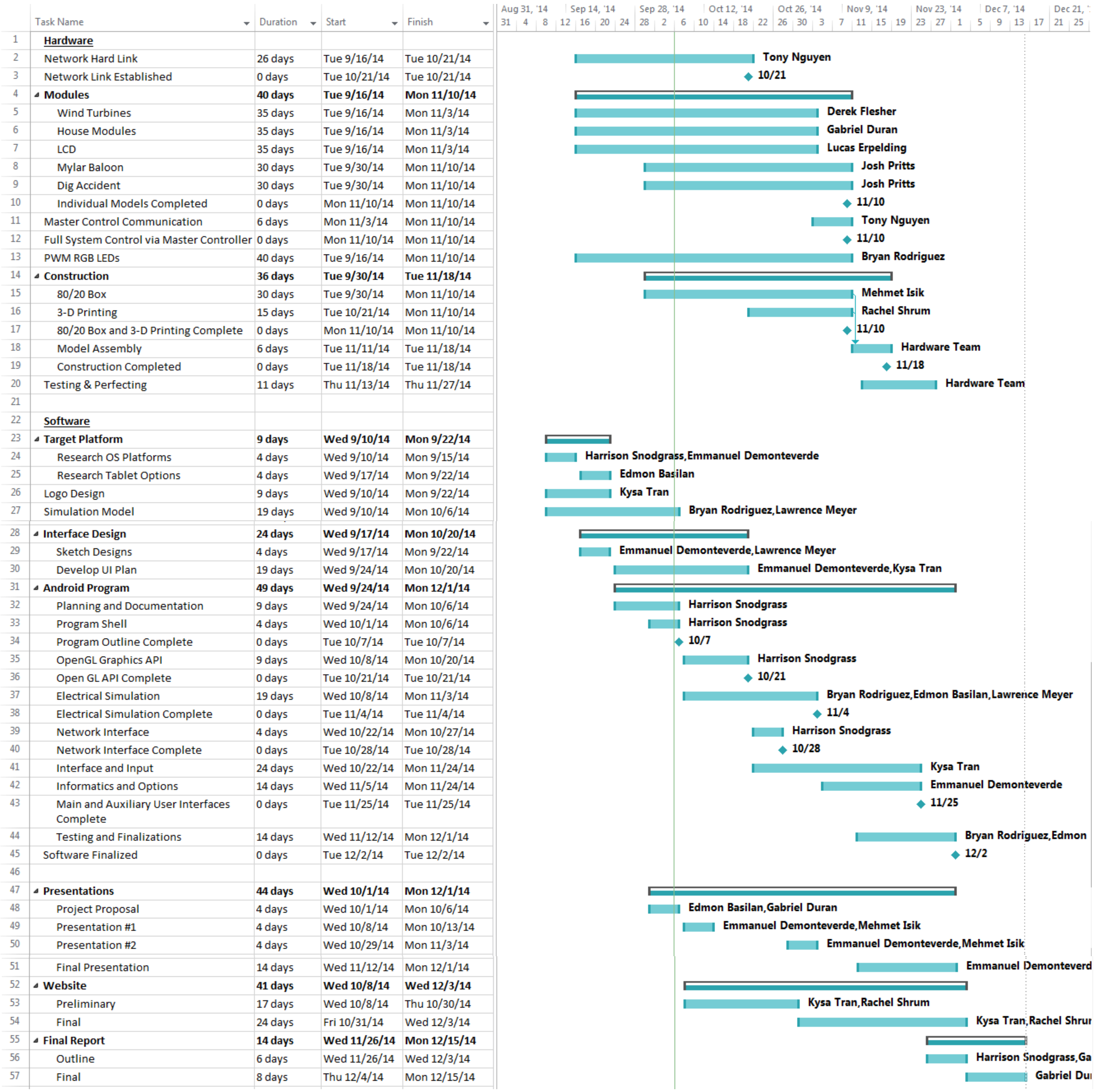
The microcontrollers used to control the different aspects of the model as well as the Wi-Fi module used to communicate with the App will be programmed using MPLabX. Landscape and buildings used for the model are designed in SolidWorks and the systems diagrams are created in MS Visio. Throughout the project our team will utilize the parts and test equipment found in the senior design lab to construct and test our systems.

Software Plan

The major code responsibilities were broken up into sections, each of which having been assigned to an individual programmer. These sections are Main, Graphics, Network, Simulation, Interface, and Informatics.

Ultimately, the code will grow and the software will fall into place as planned due to the generous amount of grace time we have built into our timeline. The code is subject to group oversight and will be edited over time. At the end of the development period, the code will be frozen and every programmer will find and fix bugs remaining in the program.

Gantt Chart



	Task Name	Duration	Start	Finish
1	Hardware			
2	Network Hard Link	26 days	Tue 9/16/14	Tue 10/21/14
3	Network Link Established	0 days	Tue 10/21/14	Tue 10/21/14
4	**Modules**	**40 days**	**Tue 9/16/14**	**Mon 11/10/14**
5	Wind Turbines	35 days	Tue 9/16/14	Mon 11/3/14
6	House Modules	35 days	Tue 9/16/14	Mon 11/3/14
7	LCD	35 days	Tue 9/16/14	Mon 11/3/14
8	Mylar Baloon	30 days	Tue 9/30/14	Mon 11/10/14
9	Dig Accident	30 days	Tue 9/30/14	Mon 11/10/14
10	Individual Models Completed	0 days	Mon 11/10/14	Mon 11/10/14
11	Master Control Communication	6 days	Mon 11/3/14	Mon 11/10/14
12	Full System Control via Master Controller	0 days	Mon 11/10/14	Mon 11/10/14
13	PWM RGB LEDs	40 days	Tue 9/16/14	Mon 11/10/14
14	**Construction**	**36 days**	**Tue 9/30/14**	**Tue 11/18/14**
15	80/20 Box	30 days	Tue 9/30/14	Mon 11/10/14
16	3-D Printing	15 days	Tue 10/21/14	Mon 11/10/14
17	80/20 Box and 3-D Printing Complete	0 days	Mon 11/10/14	Mon 11/10/14
18	Model Assembly	6 days	Tue 11/11/14	Tue 11/18/14
19	Construction Completed	0 days	Tue 11/18/14	Tue 11/18/14
20	Testing & Perfecting	11 days	Thu 11/13/14	Thu 11/27/14
21				
22	Software			
23	**Target Platform**	**9 days**	**Wed 9/10/14**	**Mon 9/22/14**
24	Research OS Platforms	4 days	Wed 9/10/14	Mon 9/15/14
25	Research Tablet Options	4 days	Wed 9/17/14	Mon 9/22/14
26	Logo Design	9 days	Wed 9/10/14	Mon 9/22/14
27	Simulation Model	19 days	Wed 9/10/14	Mon 10/6/14
28	**Interface Design**	**24 days**	**Wed 9/17/14**	**Mon 10/20/14**
29	Sketch Designs	4 days	Wed 9/17/14	Mon 9/22/14
30	Develop UI Plan	19 days	Wed 9/24/14	Mon 10/20/14
31	**Android Program**	**49 days**	**Wed 9/24/14**	**Mon 12/1/14**
32	Planning and Documentation	9 days	Wed 9/24/14	Mon 10/6/14
33	Program Shell	4 days	Wed 10/1/14	Mon 10/6/14
34	Program Outline Complete	0 days	Tue 10/7/14	Tue 10/7/14
35	OpenGL Graphics API	9 days	Wed 10/8/14	Mon 10/20/14
36	Open GL API Complete	0 days	Tue 10/21/14	Tue 10/21/14
37	Electrical Simulation	19 days	Wed 10/8/14	Mon 11/3/14
38	Electrical Simulation Complete	0 days	Tue 11/4/14	Tue 11/4/14
39	Network Interface	4 days	Wed 10/22/14	Mon 10/27/14
40	Network Interface Complete	0 days	Tue 10/28/14	Tue 10/28/14
41	Interface and Input	24 days	Wed 10/22/14	Mon 11/24/14
42	Informatics and Options	14 days	Wed 11/5/14	Mon 11/24/14
43	Main and Auxiliary User Interfaces Complete	0 days	Tue 11/25/14	Tue 11/25/14
44	Testing and Finalizations	14 days	Wed 11/12/14	Mon 12/1/14
45	Software Finalized	0 days	Tue 12/2/14	Tue 12/2/14
46				
47	**Presentations**	**44 days**	**Wed 10/1/14**	**Mon 12/1/14**
48	Project Proposal	4 days	Wed 10/1/14	Mon 10/6/14
49	Presentation #1	4 days	Wed 10/8/14	Mon 10/13/14
50	Presentation #2	4 days	Wed 10/29/14	Mon 11/3/14
51	Final Presentation	14 days	Wed 11/12/14	Mon 12/1/14
52	**Website**	**41 days**	**Wed 10/8/14**	**Wed 12/3/14**
53	Preliminary	17 days	Wed 10/8/14	Thu 10/30/14
54	Final	24 days	Fri 10/31/14	Wed 12/3/14
55	**Final Report**	**14 days**	**Wed 11/26/14**	**Mon 12/15/14**
56	Outline	6 days	Wed 11/26/14	Wed 12/3/14
57	Final	8 days	Thu 12/4/14	Mon 12/15/14

Hardware Milestones
Network Hardware Link – October 21
Responsible Party: Tony Nguyen

Individual Interactive Modules – November 3
Responsible Parties: Derek Flesher (Wind turbines)
Gabriel Duran (Houses)
Lucas Erpelding (LCD)
Josh Pritts (Mylar balloon and dig accident)
Bryan Rodriguez (RGB LEDs)

80/20 Box – November 10
Responsible Party: Mehmet Isik

3-D Printing of Landscape – November 10
Responsible Party: Rachel Shrum

Full System Control via Master Controller – November 10
Responsible Party: Tony Nguyen

Software Milestones

Program Outline – October 7
Responsible Party: Harrison Snodgrass

OpenGL API – October 21
Responsible Party: Harrison Snodgrass

Network Interface – October 28
Responsible Party: Harrison Snodgrass

Electrical Simulation – November 4
Responsible Party: Edmond Basilan

Main User Interface – November 25
Responsible Party: Kysa Tran

Auxiliary User Interface – November 25
Responsible Party: Emmanuel Demonteverde

Software Finalized – December 2
Responsible Party: Software Team

Cost Analysis

This is an estimated cost breakdown for the overall project for the various tasks needed to complete the project. This estimate does not include state tax or shipping for the various parts.

Allocated Tasks/Modules	**Cost**
Asus Tablet	$350.00
Light Animations	$872.80
Frame	$600.00
3D Models	$1000.00
Windmills	$97.02
House Modules	$166.61
Power Supply and distribution	$145.44
Communications Wi-Fi Module	$50.00
GLCD	$60.49
Various Parts for Model	$200.00
Total	$**3542.36**