Power Demand Response System

ECE 445 Design Document

Team 21
Antonio Martinez and William Widjaja
TA: Evan Widloski

1. Introduction

1.1 Problem

Stability of the grid is an increasingly difficult problem with the introduction of renewable energy sources. From a 2018 study on consumer energy management, 68% (up 3% from 2016) of residential consumers are very concerned about climate change and their carbon footprints while 7 in 10 businesses report that their customers are demanding that a certain percentage of their electricity comes from renewable sources (up 9 points from 2017) [1]. Clearly, there is a growing demand by both consumers and businesses for an increase in renewable energy sources. The problem is that these renewable energy sources, wind and solar, are not stable and are therefore difficult to predict in terms of energy output. Each new source creates another point which must be balanced against the demand. Therefore, a system of energy management must be put in place in order to maintain balance. Traditionally, energy is balanced from the perspective of the utility company.

In 2015, the California Public Utilities Commission ordered its state's investor-owned utilities to adopt time-of-use (TOU) rates by 2019 [2]. This method of energy management charges consumers more during peak demand hours and rewards consumers who limit their energy during those hours. This alleviates some of the stress put on the power grid, but it also creates expensive pricing. It is still up for debate how beneficial this TOU method is compared to other ways of managing power consumption such as tier-rates where consumers pay prices dependent on how their usage compares to overall average usage.

1.2 Solution

It is, however, possible to balance the grid strictly from the demand side while still having monetary incentivisation. TOU methods all share a dependance on the consumer to consciously and actively manage their own consumption. Our solution is to implement powerline communication between a Master Node and the Slave Nodes of each participating house. The Master Node accepts scheduling information sent by a power source, say a power plant, and outputs individual schedules to each slave node. This is illustrated in the Visual Aid section below. These devices work together to schedule the duty cycle of appliances (loads) in participating homes, such to reduce the peak load. With our functioning communication system, consumers may one day be able to opt-in to have their utilities be effectively controlled from the demand side.

Implementing demand response systems requires significant coordination of a large number of varying loads drawing power from the same grid. By developing a centralized means of communicating to loads on the grid, a utility company could more accurately predict the regulated consumer demand, lowering the total capacity required and passing the economic benefit on to the consumers who participate.

1.3 Visual Aid

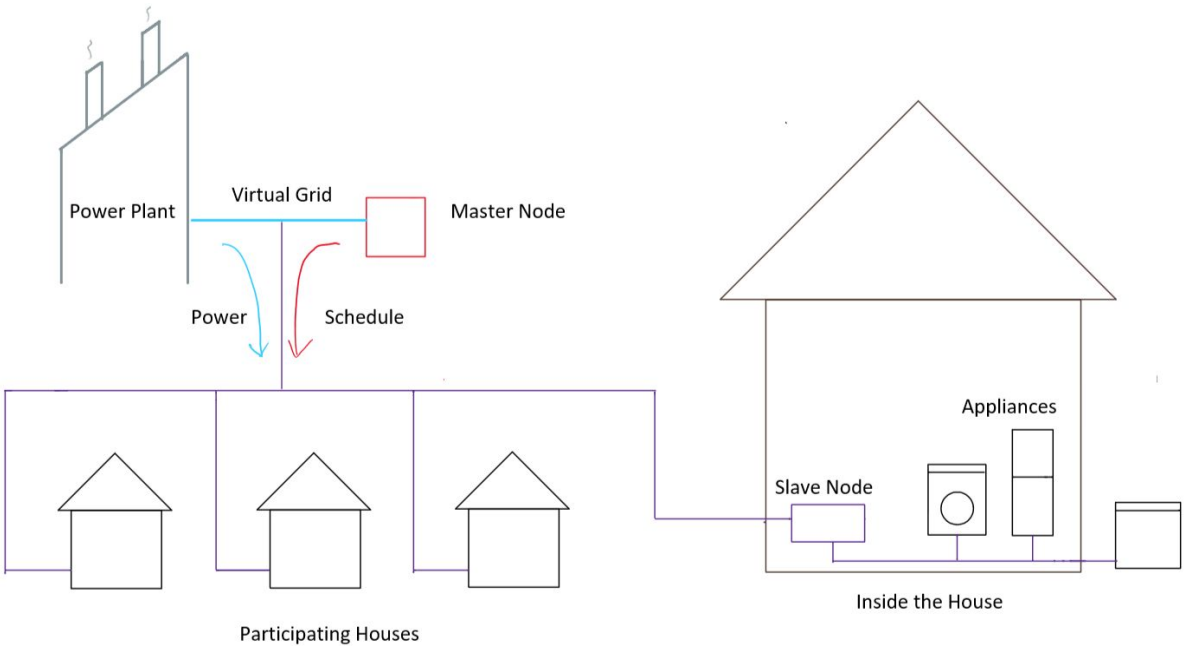

Figure 1. System Diagram

Figure 1. illustrates how we foresee the use of our system with the power grid. The power source sends power to the houses and a signal for the Master Node to interpret. Once interpreted, scheduling announcements are sent to the Slave Nodes of each participating house, and each Slave Node commands power usage of connected appliances based on the schedule. Blue lines represent power, red represents scheduling data, and purple represents both.

1.4 High-Level Requirements

1. Successfully transmit data from the power source across the powerline (the virtual grid) at a reasonable rate. An n-bit signal should be able to ride a carrier wave to a receiver on the load end within a couple of minutes.
2. Effectively dictate the on/off states of connected appliances based on schedules received from transmitted data.
3. Schedules are adjustable based on seasonal demand. For example, the demand for the Fall is different from Winter, so these seasons require different schedules.

2. Design

2.1 Block Diagram

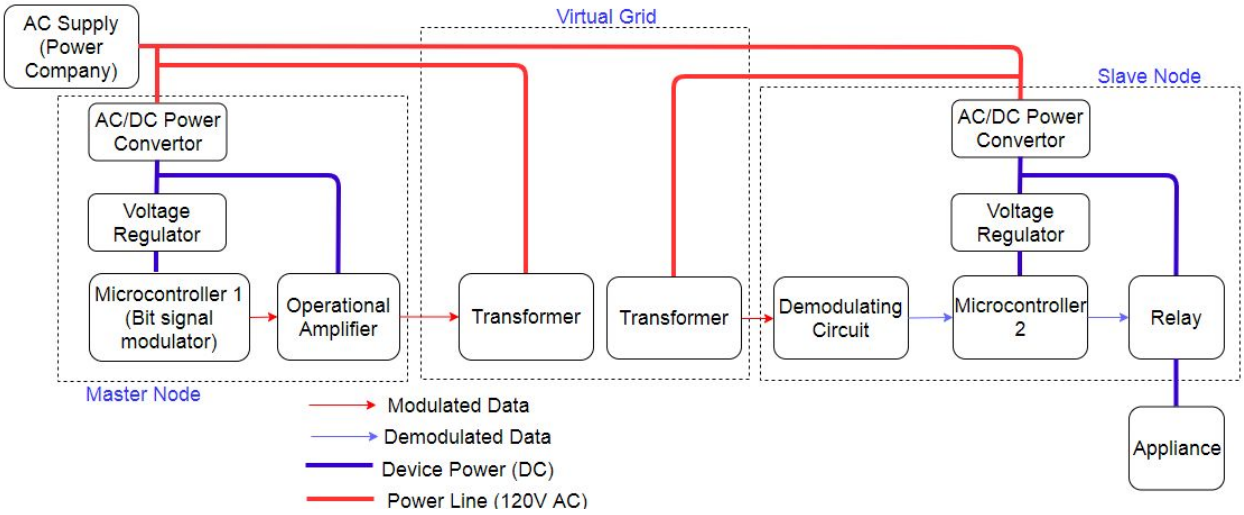

Figure 2. Block Diagram

Figure 2. Displays our three required subsystems: the Master Node, the Virtual Grid, and the Slave Node. The Master Node processes scheduling data via on-off keying and passes it to an amplifier to send it across the Virtual Grid. At the other end, the Slave Node receives the scheduling data where a microcontroller then commands connected appliances according to the schedule. New scheduling data sent to the Slave Node will appropriately change the microcontroller's command on appliances.

2.2 Master Node

The Master Node modulates the binary signal containing scheduling data for the Slave Node and amplifies the modulated signal so that it can transmit over the Virtual Grid.

2.2.1 AC/DC Power Converter

Converts the 120VAC supply to 12V DC so it can supply power for Microcontroller 1 (MC1) and the Operational Amplifier (OpAmp).

Requirements	Verification
1. Converts 120 VAC to 12 V DC 2. Successfully supplies voltage to MC1 and OpAmp	1. Measure input and output voltages using a voltmeter 2. LED indicating MC1 is on is lit and measured output of OpAmp is

	amplified by desired value

2.2.2 Voltage Regulator

Our MC1 operates at 1.8 - 5.5V, so we must regulate the 12V from the AC/DC Converter down to 5V. The Voltage Regulator ensures our MC1 is getting proper supply.

Requirements	Verification
1. Output of Voltage Regulator is not above 5V	1. Measure output using voltmeter

2.2.3 Microcontroller 1 (Bit signal modulator)

Microcontroller 1 has several functions; it must generate a bit signal which holds the seasonal scheduling data that the Slave Node will execute, it modulates the bit signal with its on-chip RC oscillator signal, and finally outputs the modulated signal to the OpAmp so that it can be delivered down the Virtual Grid.

We use on-off keying (OOK), the simplest form of amplitude shift keying modulation. The process of OOK has a high frequency carrier signal (the internal oscillator signal) modulated with a square signal (the bit signal), creating a modulated signal that represents the original carrier signal whenever the binary signal is 1, and a flatline signal whenever the square signal is 0. An example is illustrated the Figure 3.

We must establish a period T based on the frequency that represents 1 bit. This is to differentiate the carrier wave oscillating from -1 to 1 from the flatline portions of the modulated wave. So say peak-to-peak oscillation takes 1s; we establish 1s = 1bit. Now say the 3 initial flatline portions are 2s, 1s and 3s long respectively, then our modulated signal represents 00111011000. Now that our digital code is represented in analog form, it can be transmitted through the virtual grid.

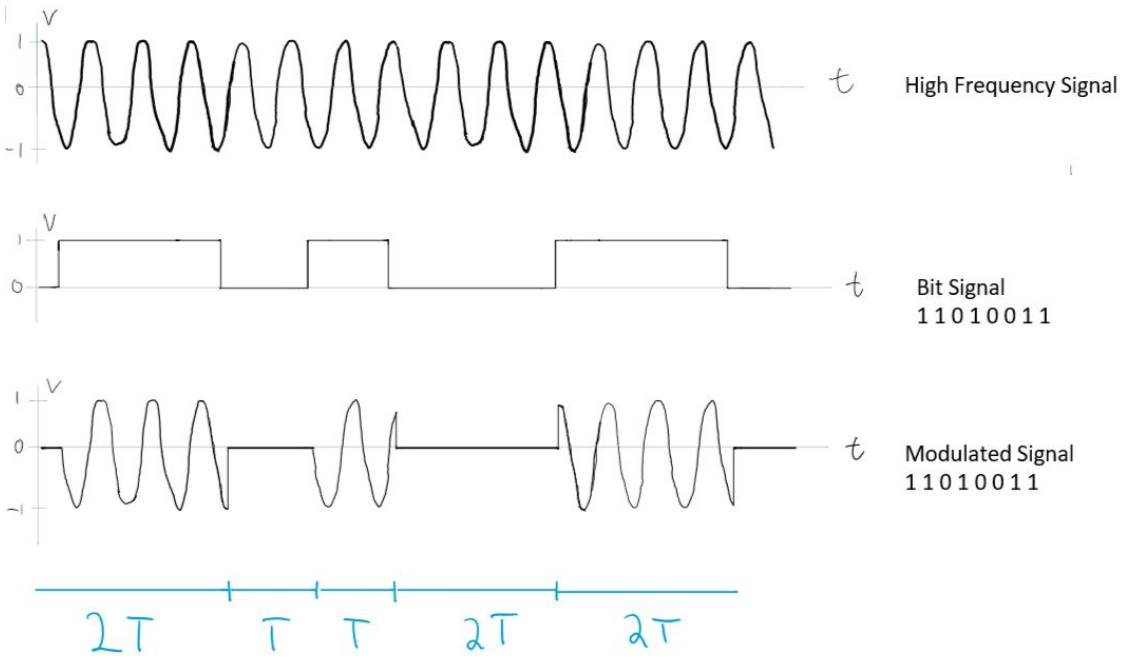

Figure 3. OOK Visual Representation

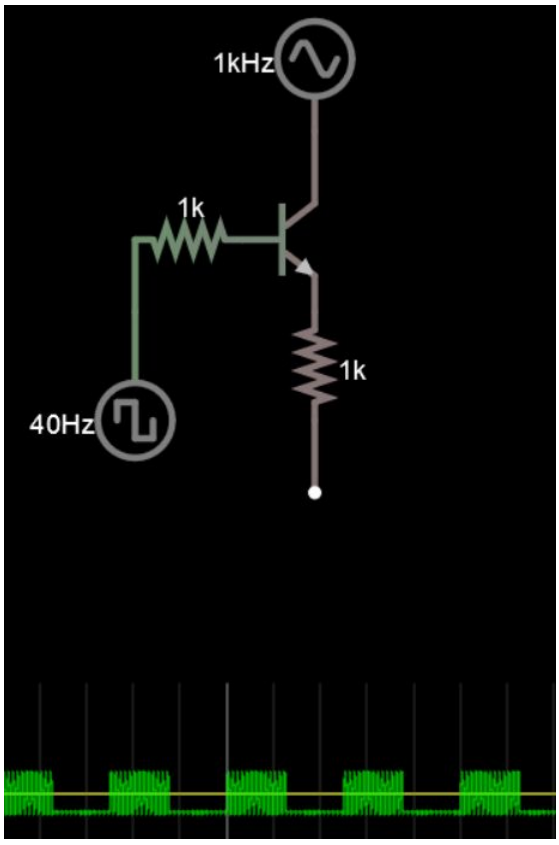

Figure 4. Falstad Simulation of OOK Circuit with Scoped Output

In Figure 4. we simulated the modulation process that our MC1 would do. The non-zero input by the digital signal (40Hz square wave) will activate the transistor so that the carrier signal (1kHz AC source) is the output. For a zero input by the digital signal, there is a ~-1V output. Ideally, the output would be zero. If we used this circuit instead of the MC1, the output of the emitter would connect to the OpAmp to be amplified. It's worth noting that the bit signal should run at a frequency lower than the oscillating signal's frequency. This is to retain the shape of the oscillating signal during active high periods of the bit signal. This will be further discussed in the Tolerance Analysis.

Requirements	Verification
1. Generates bit signal at frequency much lower than oscillating signal (i.e. 60Hz : 1kHz) 2. Modulates bit signal with on chip oscillating signal and outputs modulated signal with correct frequency (i.e. 1kHz) 3. Changes bit signal based on seasonal demand	1. Program MC1 to output bit signal so we can display and measure it's frequency using an oscilloscope while also confirming the correct sequence of bit signal. 2. Record output signal on oscilloscope and confirm its frequency while also verifying correct shape. 3. Verify change in bit signal after an intended period using oscilloscope

2.2.4 Operational Amplifier

Because the modulated signal is 1VAC, its small voltage value makes it susceptible to noise while being transmitted down the power line. A larger valued signal will also have a better defined envelope once demodulated. For these reasons, we must amplify the modulated signal by x10. Figures 5a and 5b demonstrate the need for this OpAmp.
In Figure 5a, we demonstrate how the x10 amplified modulated signal looks after demodulation. Its envelope is much better defined than the shape shown in Figure 5b where there was a x1 amplification. Note that the amplified signal's envelope has a better discharge shape compared to the rest of the envelope. This will be discussed further in the Tolerance Analysis.

Requirements	Verification
1. OpAmp successfully amplifies by x10	1. Use oscilloscope to measure input and output of OpAmp. 1V going in should have ~10V going out.

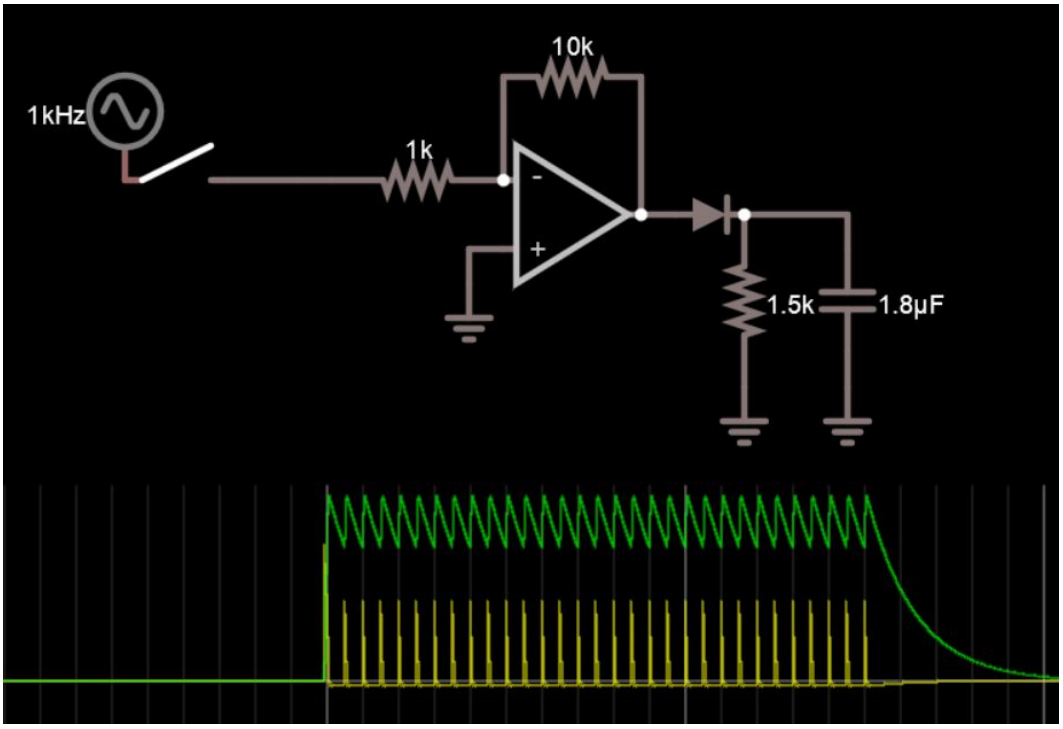

Figure 5a. Demodulation of x10 Amplified Modulation Signal

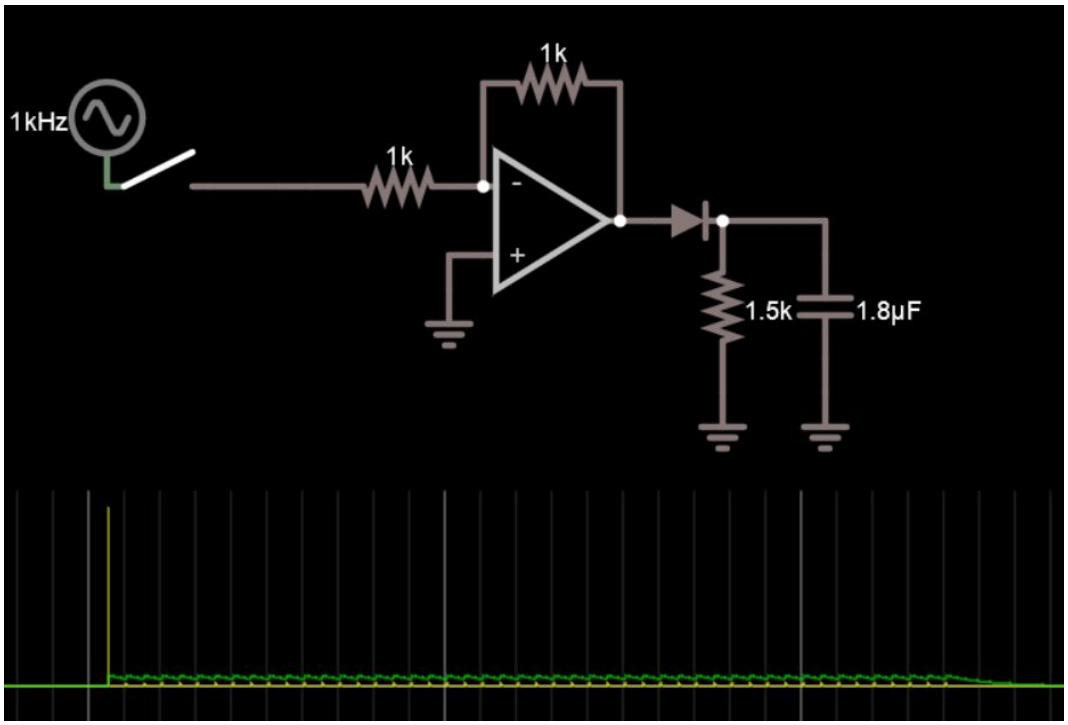

Figure 5b. Demodulation of x1 Amplified Modulation Signal

2.3 Virtual Grid

The path which our signals are traveling on. For our design we will treat this electrically as a wire with a very small resistance.

2.3.1 Transformers: These transformers are stepup/stepdown isolating transformers. They mimic the real transformers used in transmission lines to decrease line loss and to protect the rest of the circuit.

Requirements	Verification
1. All signals made by the system must be under 500 kHz by U.S. law. No radio waves in free space if unshielded wiring is used.	1. Measure signals at each electrical node and ensure there are no frequency components above 500 kHz

2.4 Slave Node

This subsystem effectively dictates the duty cycles of connected appliances according to the schedule sent by the Master Node over the Virtual Grid. It must first process the Master Node's modulated signal then control the duty cycle of its appliances.

2.4.1

Demodulating Circuit: Acts as an asynchronous demodulator to convert the input analog signal into a digital output readable by the microcontroller. As shown in Figure 6., this module consists of a half-wave rectifier to deliver a positive half output into a low-pass filter. This filter delivers the envelope of the signal which is shaped more into a digital signal. Finally, the envelope is sent to a comparator to finalize the digital signal. For our system, the Demodulator Circuit consists of the rectifier and LPF while the comparator is in the receiving microcontroller.

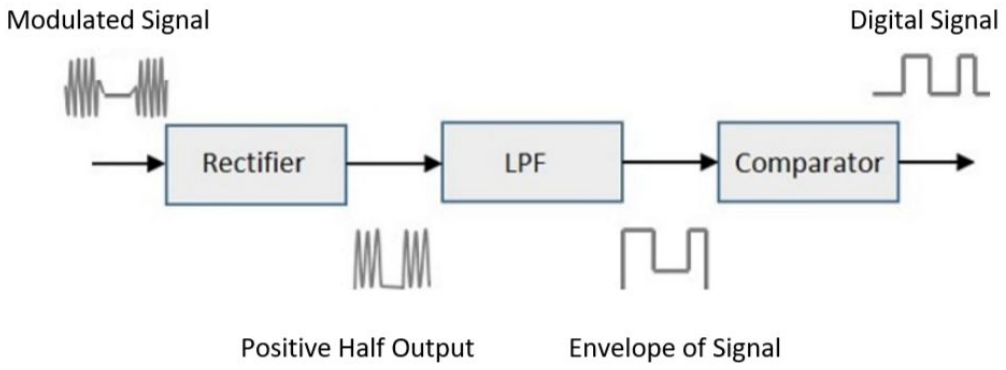

Figure 6. Demodulator Diagram

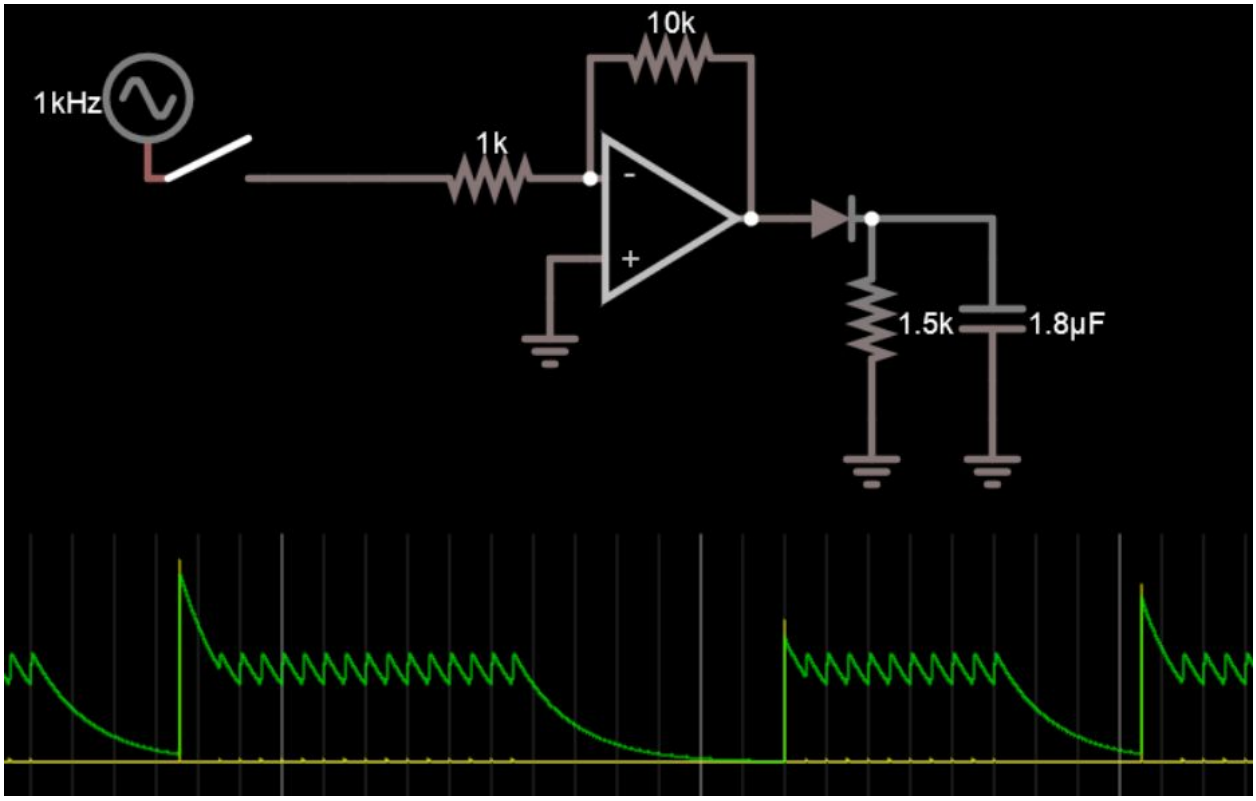

Figure 7. Falstad Simulation of Demodulator

We can see from Figure 7. The exact envelope we'll get as the 1kHz signal is toggled (to act as a modulated signal). For the receiving microcontroller to correctly read the demodulated signal, parameters of this circuit must be carefully chosen. Depending on the resistance, capacitance, and frequency of the signal, the shape can have bigger or smaller jagged peaks ($Vripple = Iload/fc$), longer or smaller capacitance discharge time ($Vcap = Vsource\, e^{-t/RC}$), and faster or smaller bit rate. By varying the resistance and capacitance of the RC filter, we've chosen this shape because the jagged peaks aren't too big and the discharge time isn't too long. The relationship between the envelope's shape and bit rate will also be further discussed in the Tolerance Analysis.

Requirements	Verification
1. Must decode the information from the transmitter correctly, at least in an ideal situation where the transmitted signal is perfect/directly in communication with the receiver. For example, if 1011 is transmitted, 1011 should be received.	1. Scope the demodulator output and the bit signal of MC1. Their shapes should be within 10% of rise and fall time delay. For example, if bit signal rises at T = 1s and falls at T=2s to represent a digital 1, the demodulated circuit should rise and fall in 1(+-0.1)s to represent digital 1. This gives ample room for discharge time.

2.4.2 Microcontroller 2

Essentially opens or closes the relay for the load based on initial binary sequence representing schedule commands being properly transmitted and received.

The data which is being sent through the virtual grid is an array of arrays. Since this is akin to a “broadcast” on the grid, all device schedules are actually available in this array. The following pseudocode illustrates this:

```
allSchedules = [A, B, C, D, E … ] //This is the array with all schedules

B = [ deviceId = “B”, changeSchedule = true, newSchedule = [12pm: on, 1pm: off, default: off], currentTime = 12pm ]
```

Code Sample 1: Array “B” is an example of one of the subarrays. This subarray contains scheduling information for a load with serial identification “B.” Only the device with matching serial ID = “B” will follow the instructions set by array B. In this specific array, the device B will change its schedule and receive power from 12pm to 1pm, where it will then turn off and stay off until a new instruction arrives.

```
#include <stdio.h>
#include <stdlib.h>

// struct codeWord with 3 fields
int lengt = 64
struct codeWord
{
    int id;
    char firstName[lengt];
    char birthday[lengt];
};

// Driver program
int main ()
{
    FILE *infile;
    struct codeWord input;

    // Open codeWord.dat for reading
    infile = fopen ("codeWord.dat", "r");

    if (infile == NULL)
    {
        fprintf(stderr, "\nError opening file\n");
```

```
        exit (1);
    }

    // read file contents until end of file reached
    while(fread(&input, sizeof(struct codeWord), 1, infile))
        printf ("id = %d name = %s %s\n", input.id,
        input.firstName, input.birthday);

    fclose (infile);
    return 0;
}
```

Code Sample 2: C code which reads from a struct called “codeWord.” This struct will store the actual data. The length of each codeWord is determined by the “lengt” variable and is tentatively 64 characters. This is 64 bytes. On the final version, a buffer method will constantly search for a special start character. When the start character is detected, that means a codeWord was received and the next 64 bytes will be stored either in another variable or to the memory of the microcontroller

Requirements	Verification
1. Must store correct bit representations as intended by transmitter/receiver and properly control the relay switch with those signals.	1. Send a test input to the microcontroller and then view the output through 4 LEDs or a console. The 4 LEDs represent the next 4 hours of operation and LED on = power and LED off = no power. The console command would just output a 4 bit array where 1 = power and 0 = no power

2.4.3 Relay

The switch that opens or closes depending on the commands of the microcontroller so that a load (appliance) is connected to power. This is also closer to a “data sheet” parameter, but worth mentioning since it controls power to the appliance, the final step and goal of the data word sent.

Requirements	Verification
1. Must open and close a circuit with no more than 5s delay.	1. Send a test input and measure the time it takes for the relay to react

2.5 Tolerance Analysis and Feasibility

First we must return to our modulation procedure to analyze why the bit signal must have a lower frequency than its carrier signal. In Figure 8. The frequency of the carrier frequency is toggled from 40Hz, 100Hz, 500Hz, and 1000Hz. Clearly, the closer the frequency is to the square signal, the less the modulated signal retains the sinusoidal shape of the carrier frequency. It is for this reason the carrier frequency should be much larger than the bit signal.

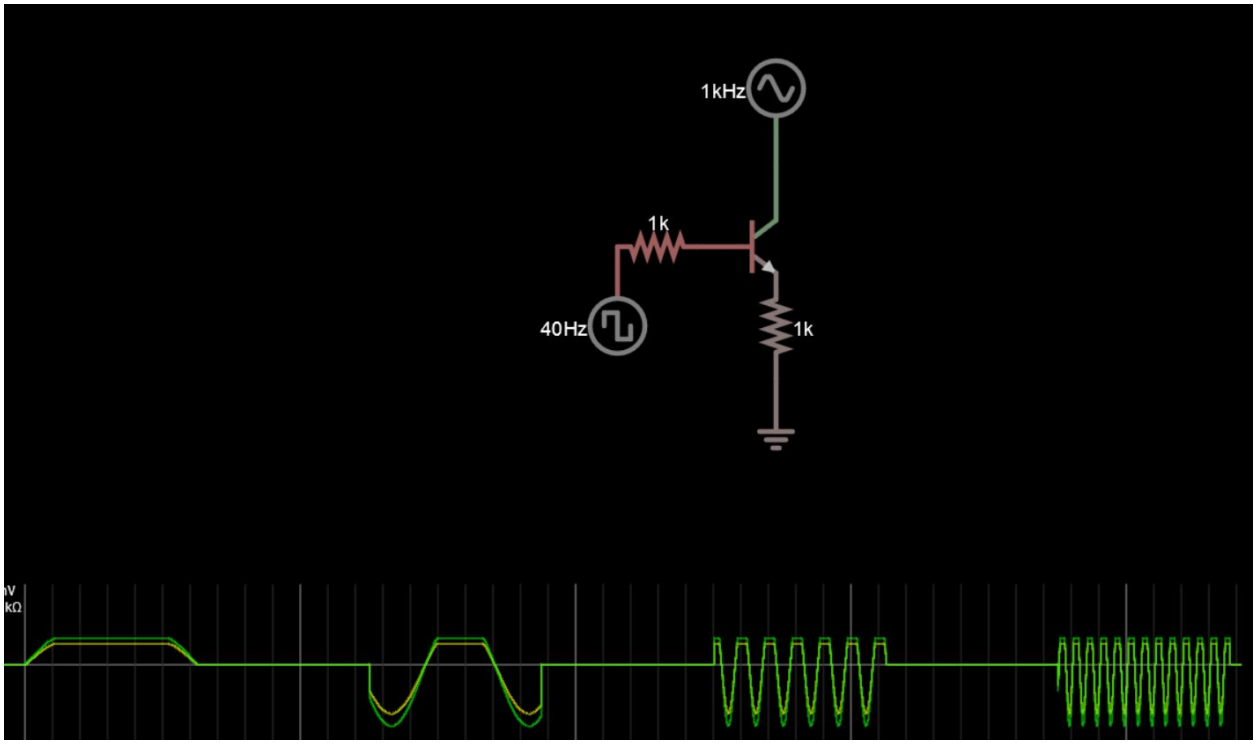

Figure 8. Modulation with Varying Carrier Signal Frequency

.

The bit rate and the resulting data transmitted depend greatly on the widths of the logical 1s and 0s within the modulated signal across the virtual grid. When the signal is demodulated and read by the microcontroller at a slave node, it must be able to handle variances in the bytes incoming, otherwise the data read will not match the data sent.

The discharge time depends on the high analog voltage. In turn, that voltage is proportional to the RC time constant. If the resistance (of the 1500 ohm resistor in the Falstad simulations of the demodulator) is low, the discharge time decreases, but the voltage ripple increases. If the resistance is increased, the voltage ripple is low, but the discharge time increases, lowering the overall data rate.

We want a speed of at least 32 bits per second. A single message consisting of 64 bytes would take about 2.1 minutes to send. Since the duty cycles we aim to change are on the order of 30-60 minutes, a few minutes is reasonable speed for sending messages to change them.

Figure 9. A simulation of a high resistance value. Discharge time increases, resulting in longer width digital 1s. This could be compensated for with additional “dead time” if identical length 1s and 0s are needed

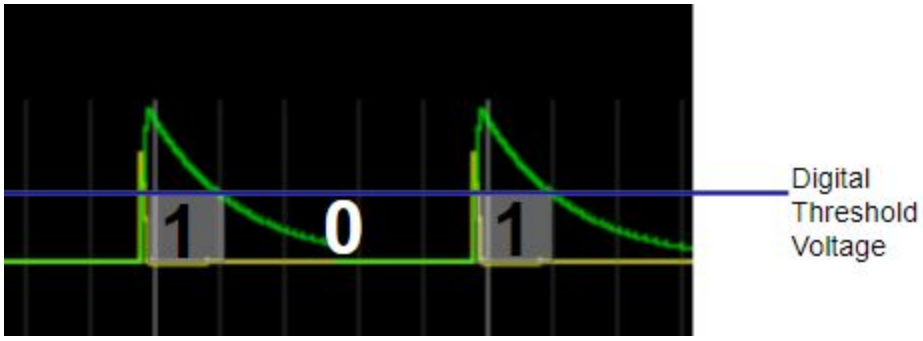

Figure 10. A simulation of a low resistance value. This results in very short duration logical 1s.

We estimate a ripple over 60 mV would be too high relative to a digital threshold of 0.5 V, where anything above 0.5 V would be digital 1 and anything below would be digital 0. We will choose a resistor high enough to smooth the ripple below 60 mV, but If the 1s are more than 10% longer than the 0s, or there are less than 32 discernible bits within a 1 second interval, then the resistance would be too high.

Now we take a closer look at Figure 5. An amplified signal has a clear difference between high analog voltage and digital threshold voltage where a non-amplified signal has too small of a difference. This is shown in Figure 10. Why not amplify by something greater than x10? Again, we look at discharge time. More voltage requires longer discharge time. This is our justification of choosing x10 amplification.

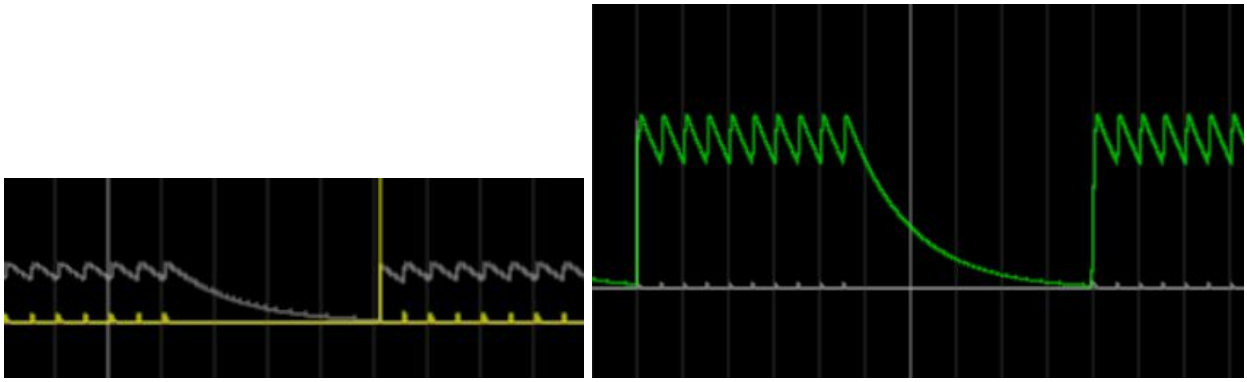

Figure 11. Envelope of Non-Amplified (left) and Amplified (right) Demodulated Signals

Next, in order to justify the benefit of the system, it is helpful to quantify the total load required in the case when cycles are random and when the cycles are complementary. The following figures show the difference in coordinated and random loads.

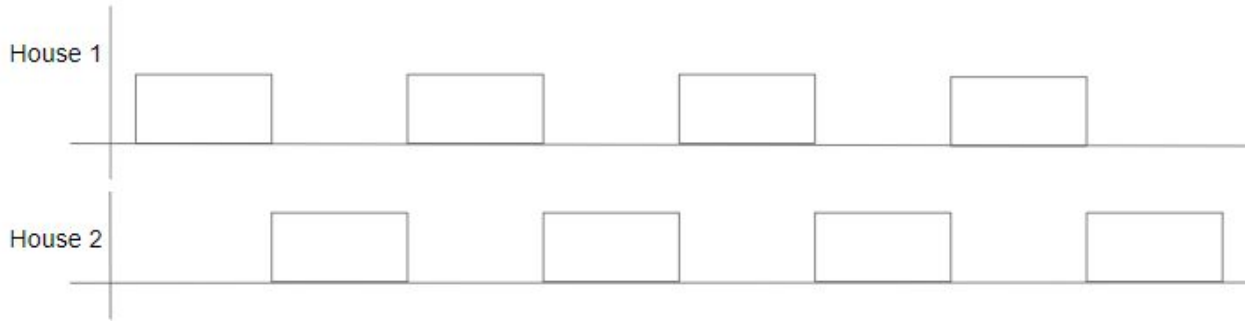

Figure 12. Controlled load cycles of two houses. The horizontal axis is time, the vertical axis is power used per house.

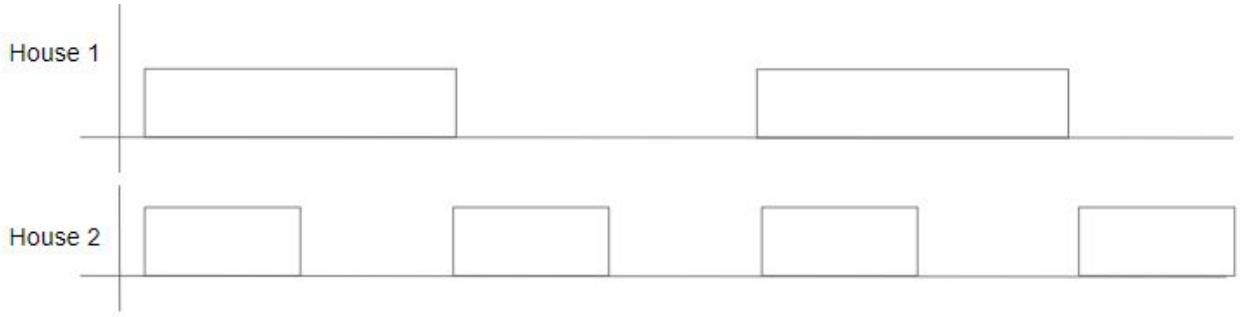

Figure 13. Example of random load cycles of two houses. The horizontal axis is time, the vertical axis is power used per house.

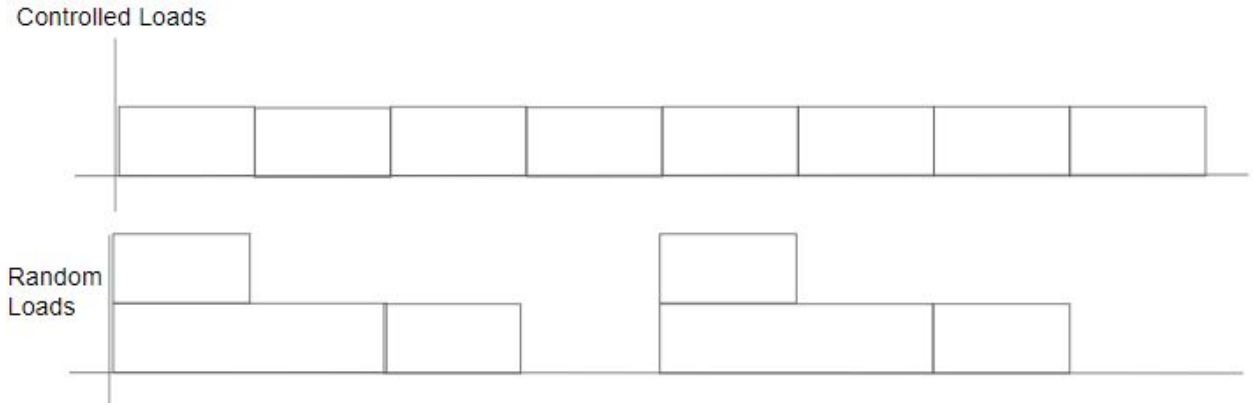

Figure 14. Total loads seen from the perspective of the power plant. The energy used is the same in terms of area, but there is a cost savings derived from having a lower peak power requirement.

2.6 Cost Analysis:

2.6.1 Estimated Total Labor Cost

Engineer	Hourly Rate	Hours (12 hours/week x 12 weeks)	Hourly Rate x Hours x 2.5
Antonio Martinez	$40.00	144	$14,400
William Widjaja	$40.00	144	$14,400
			Total: $28,800

2.6.2 Estimated Total Parts Cost

Description (hyperlinked)	Manufacturer	Part Number	Quantity	Unit Cost	Total Cost (prices may vary on other days)
Assorted Resistors, Capacitors, Wires	Texas Instruments	Various	5 of each from 10, 20, 50, 100, 1k, 10k and 1M	$0.40	$14.00
Isolating Transformer (N-48X)	Triad Magnetics	237-1856-ND	2	$14.62	$29.24
Step Up/ Step Down Transformer(3 FD-210)	Tamura	MT2096-ND	2	$5.73	$11.46
1N4004 Diodes	Diodes Incorporated	1N4004DICT-ND	6	$0.18	$1.08
Arduino Uno Rev3	Mouser Electronics	8058333490090	1	$22.00	$22.00
Microcontroller	Microchip Technology	ATMEGA328-PU-ND	2	$1.96	$3.92
Operational Amplifier	Mouser Electronics	LM741	3	$1.00	$3.00
555 Timer Chip	Texas Instruments	LM555CMM/NOPB	2	$1.00	$2.00
					Total: $86.70

Grand Total: Labor Costs + Parts Cost = $28,800 + $86.70 = **$28,886.70**

2.7 Schedule:

Week of	Antonio	William
9/30	Design Document, Simulate Circuit	Design Document, Order Parts
10/7	Acquire Parts, Create and test breadboard circuit of modulator	Acquire Parts, Create and test breadboard circuit of bit generation
10/14	Design modulator and demodulator circuits on EagleCAD. Order PCB	Data generation circuit created and program written for sending bits. Order PCB.
10/21	Acquire and test PCB. Revise if necessary. In either case, send another order for more nodes.	Combine the modulating PCBs and the data generation. Begin testing with real DC loads. Test at least one DC load.
10/28	Test with multiple nodes. Last round of PCBs submitted to test multiple nodes, just in case they are not passing tests	Test with multiple nodes. Last round of PCBs submitted to test multiple nodes, just in case they are not passing tests
11/4	Fix any PCB issues that may have come up	Software encoding/decoding and scheduling algorithms complete
11/11	Continue fixing PCB issues, mount PCBs, help William	Fix any software issues, else, prepare for Mock Demo
11/18	Prepare for Mock Demo	Prepare for Mock Demo
11/25	Work on Final Paper, Resolve Remaining Issues	Work on Final Paper, Resolve Remaining Issues
12/2	Final Demo and Mock Presentation	Final Demo and Mock Presentation
12/9	Final Presentation and Final Paper	Final Presentation and Final Paper

3. Ethics and Safety

Due to the nature of our project as a broad communication system, there is a need for proper security of the information being sent. Our project is designed to eventually help dictate the usage of compliant consumers, so it is our responsibility to come up with a secure method of sending and receiving data and being transparent with our understanding of how secure the system is.
One specific consideration is that the data being sent contains scheduling information for everyone's devices. For real applications, it would be important to implement an encryption protocol, such as public key cryptography, where the private key is safely stored on the hardware of a load device. This would secure the data being sent and ensure that each load only has access to its own scheduling data. The public key encryption would need to occur before the data enters the virtual grid.

In compliance to the IEEE Code of Ethics and the ACM Code of Ethics and Professional Conduct [3][4], we will hold the security and safety of users paramount in our development of this project to exemplify the specific codes 1.1,1.3, 1.6, and 1.7. Note that we do not restrict ourselves to these only these codes but are stating that these codes are more relevant to the nature of our project. We do not foresee any ethical dilemmas created by our project. The system is intended to be used by consumers properly educated of what our communication system does and how it is intended to control their consumption. It will not be marketed or sold to consumers who would suffer compromises to their health from the rate limited use of electricity, such as a person on hospice.

Lastly, we will be careful to follow all lab procedures described by the ECE 445 guidelines and will not hesitate to ask questions or express concerns to faculty and classmates during the process of building this project. As partners, we will hold each other accountable to follow all codes of ethics especially in holding high standards of competence, conduct, and ethical practice, knowing and respecting existing rules pertaining to laws in the power and communications industry, and assisting each other during this project development, and to support each other in following these codes.

References

[1] Deloitte. ‘Deloitte Resources 2018 Study’, 2018. [Online]. Available: https://www2.deloitte.com/content/dam/insights/us/articles/4568_Resource-survey-2018/DI_Deloitte-Resources-2018-survey.pdf

[2] H. Trabish. ‘An emerging push for time-of-use rates sparks new debate about customer and grid impacts’ *UtilityDive’*, 2019. [Online]. Available: https://www.utilitydive.com/news/an-emerging-push-for-time-of-use-rates-sparks-new-debates-about-customer-an/545009/

[3] Ieee.org. ‘IEEE IEEE Code of Ethics’, 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html.

[4] acm.org. ‘ACM Code of Ethics and Professional Conduct’, 2018. [Online] Available: https://www.acm.org/code-of-ethics