

POWER LINE COMMUNICATION

By

Antonio Martinez

William Widjaja

Final Report for ECE 445, Senior Design, Fall 2019

TA: Evan Widloski

11 December 2019

Project No. 21

Abstract

The purpose behind this project is to schedule power usage of home appliances to minimize a neighborhood's peak load. By utilizing the existing powerline infrastructure as a means of sending scheduling data, direct communication between the electric utility and the homes they supply can be established. Our project simplifies real world power lines as two wires connecting the supply-side transmitting module (Master) and the demand-side receiving module (Slave or Follower). We focused on the scheduling data generation, powerline injection and extraction of the data, and how the data is used to dictate connected appliances. Our project proved to successfully generate, send, and receive this data, however there were some issues in reading the data. Ultimately, we proved the idea could work, and realized how it could be improved upon in the future.

Table of Contents

1. Introduction	4
1.1 The Problem	4
1.2 Our Solution	4
1.3 High Level Requirements	5
1.4 Visual Aid	5
1.5 Signal Path Block Diagram	6
2 Design	6
2.1 Design Procedure	6
2.2 Simulations and Equations	7
2.2.1 Modulator	7
2.2.2 Demodulator	8
2.3 Master Transmitter Circuit Components	10
2.4 Virtual Grid Components	10
2.5 Follower Node Circuit Components	11
2.6 Microcontroller Code	11
2.6.1 Microcontroller Master Scheduling Code	11
2.6.2 Microcontroller Follower Code	12
2.6.3 Microcontroller PWM Code	13
2.7 Relay and Appliances	13
3 Signal Analysis	14
3.1 Master Node Verification	14
3.1.1 Microcontroller 1 Signal Generation Verification	14
3.1.2 Modulation Verification	14
3.1.3 Powerline Injection and Data Extraction Verification	15
3.1.4 Demodulation Verification	15
3.1.5 Relay Control Verification	16

3.1.6 Single Master Writer to Multiple Follower Reader Verification	16
4. Cost Analysis	16
4.1 Labor	17
4.2 Parts	17
4.3 Schedule	18
5. Conclusion	19
5.1 Accomplishments	19
5.2 Uncertainties	19
5.3 Ethical considerations	19
5.4 Future work	19
References	21
Appendix A Requirement and Verification Table	21

1. Introduction

1.1 The Problem

Large and unpredictable peak loads on the power grid cause the grid to generate more average power than used, therefore decreasing efficiency and increasing costs of electricity. This has been a widely known issue, but currently the power grid is at a larger risk of being inefficient with increasing usage of renewable resources. 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 sources are 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 Our Solution

It is, however, possible to balance the grid strictly from the demand side while still having monetary incentives for the end consumers. TOU methods all share a dependence on the consumer to consciously and actively manage their own consumption. Our solution is to implement powerline communication between a Master node emanating from the power company and the Follower nodes of each participating house. The Master node can be programmed to send scheduling data across the powerline to Follower nodes that execute the schedules by controlling connected appliances. This is illustrated visually in Figures 1 and 2. These devices are scheduled simultaneously in participating homes to reduce the peak load.

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 via power line injection, a utility company could more accurately predict the regulated consumer demand, lower the total peak power required, minimize time when demand is lower than supply, and pass economic benefits onto participating consumers.

1.3 High Level Requirements

1. Successfully generate scheduling data and transmit it across the powerline. 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.

1.4 Visual Aid

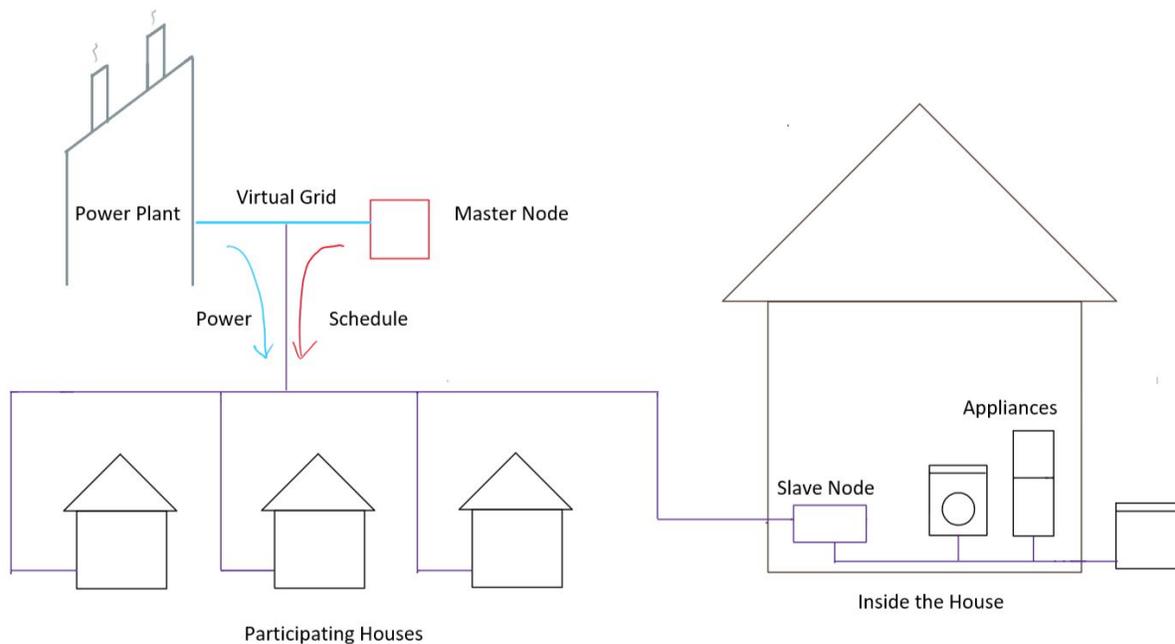


Figure 1. Real World System Diagram

Figure 1 demonstrates how we envision our system working in the real world as intended. The power company sends power across the powerline to its Master node and the Follower Nodes of each participating house. The Master node generates its programmed signal and injects it onto the powerline for Follower nodes to receive. Once received, the Follower node will then dictate connected appliances according to the schedule. Blue lines represent power, red represents scheduling data, and purple represents both.

1.5 Signal Path Block Diagram

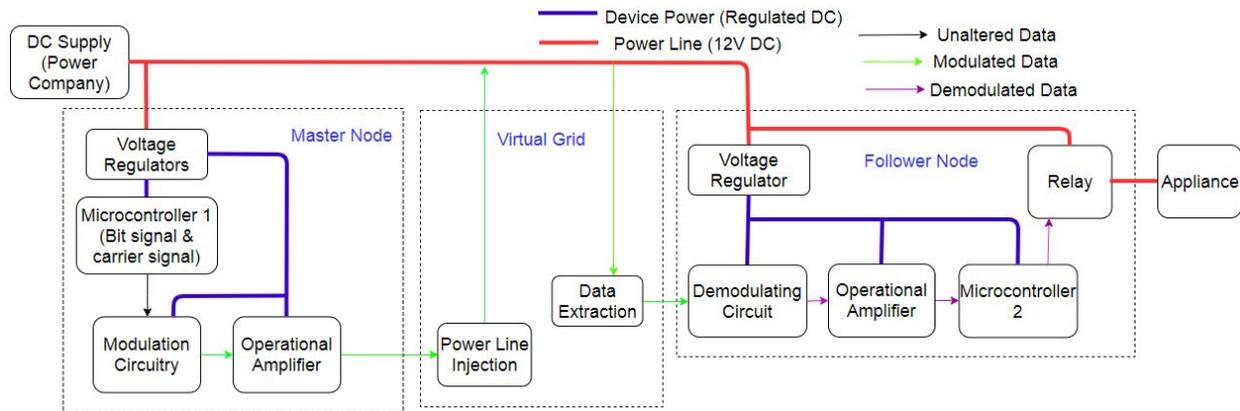


Figure 2. Block Diagram

This block diagram shows the path of power and our scheduling data as they travel to our three subsystems: Master node, Virtual Grid, and Follower node. Inside the Master node, Microcontroller 1 (MC1) generates preprogrammed schedule data as a digital bit signal and a high frequency carrier signal (HFC). These signals are modulated in the Modulation Circuitry and amplified so that they are prepared to traverse onto the Virtual Grid. This subsystem is a simulation of the real powerline connected Master and Follower. The Power Line Injection and Data Extraction blocks consist of inductors and capacitors that allow the signal to travel on and be received from powerline. Once extracted, the scheduling data must go through demodulation (redigitization) and amplification so that Microcontroller 2 (MC2) may read it. Upon interpreting the data, MC2 then governs an appliance through a relay. Note that voltage regulators also exist to limit the power supplied to all existing microcontrollers and amplifiers, for they do not operate at high voltages.

2 Design

2.1 Design Procedure

The scope of our project has altered since our initial inception of the Power Demand Response System. Specifically, the concentration of getting power supply directly from the powerline shifted away as we realized the bulk of this project has more to do with powerline communication. We realized this early on as we conducted simulations and created breadboard circuitry. Getting power directly from the powerline was not as important because it simply required a couple more circuit elements whereas the process of power line communication is a necessary key component, designed to be easily adjustable depending on how it is connected to the real grid.

Since our project revolves around signal manipulation via circuitry, our first step in the design process was to create simulations of circuitry that generated and manipulated signals that represent our bit signal. This allowed us to easily scope our signals after each step of simulation. Once we gathered enough information to generally understand the signal manipulation, we created breadboard circuitry as a precursor for the PCB. It was at the breadboard stage where we began using equations to get specific

values of necessary components. Finally, once the parameters of components were chosen, we designed the PCB, populated and tested it, and made any necessary changes.

2.2 Simulations and Equations

As stated above, simulations gave us better inside the signal manipulation we planned on doing. Because our projected relied on modulation and demodulation to work, the major risks of the project are the modulating and demodulating circuit components. These portions were simulated using Falstad online software.

2.2.1 Modulator

The modulating circuit uses the on-off keying shown in Figure 3. the bit signal and HFC are essentially multiplied to produce the modulated signal. On-off shift keying is realized by a bipolar junction transistors (BJT). As seen in Figure 3, B1 receives DC supply at its collector, the bit signal at the base, and its emitter is grounded. After this step our bit signal is inverted. Then at B2, the HFC goes to the collector, the inverted bit signal is delivered to the base, and the emitter is grounded. When the bit signal is 0, 0V is applied at the base making the modulated signal output also 0. When the bit signal is 1, the modulated signal output is the HFC. This is how we implement on-off keying. Values here are not necessarily important, except for the HFC's frequency compared to the bit signal. To get a full or multiple periods of the HFC present during bit signal is 1, its period has to be shorter than the bit signal's period or frequencies

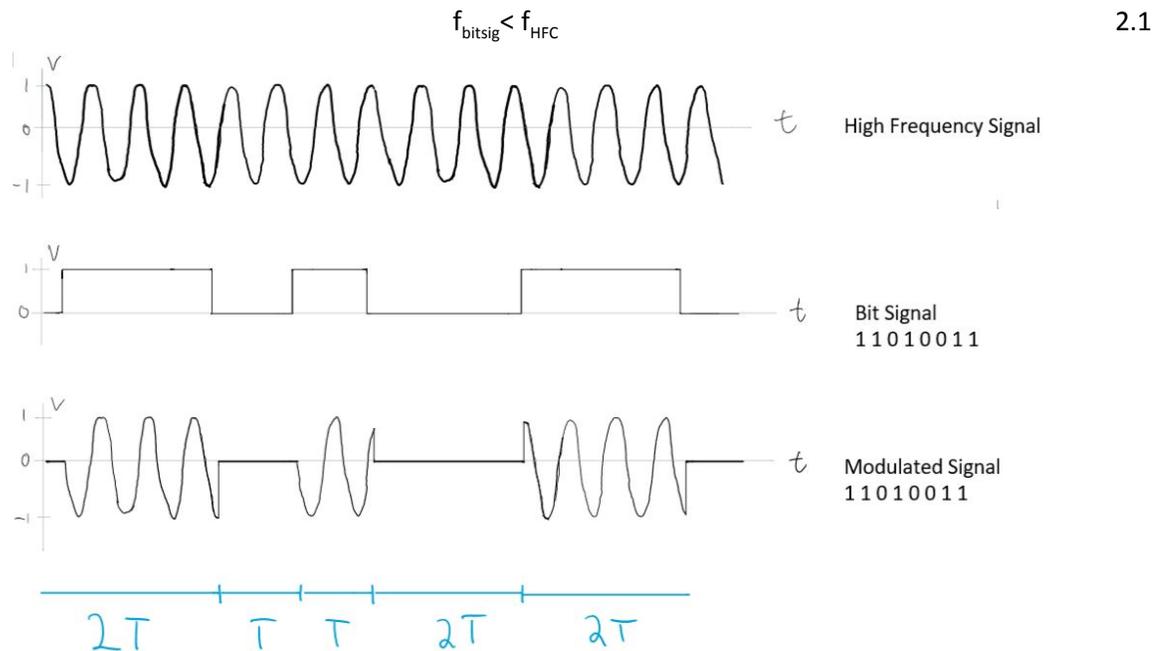


Figure 3. On-Off Keying Visualization

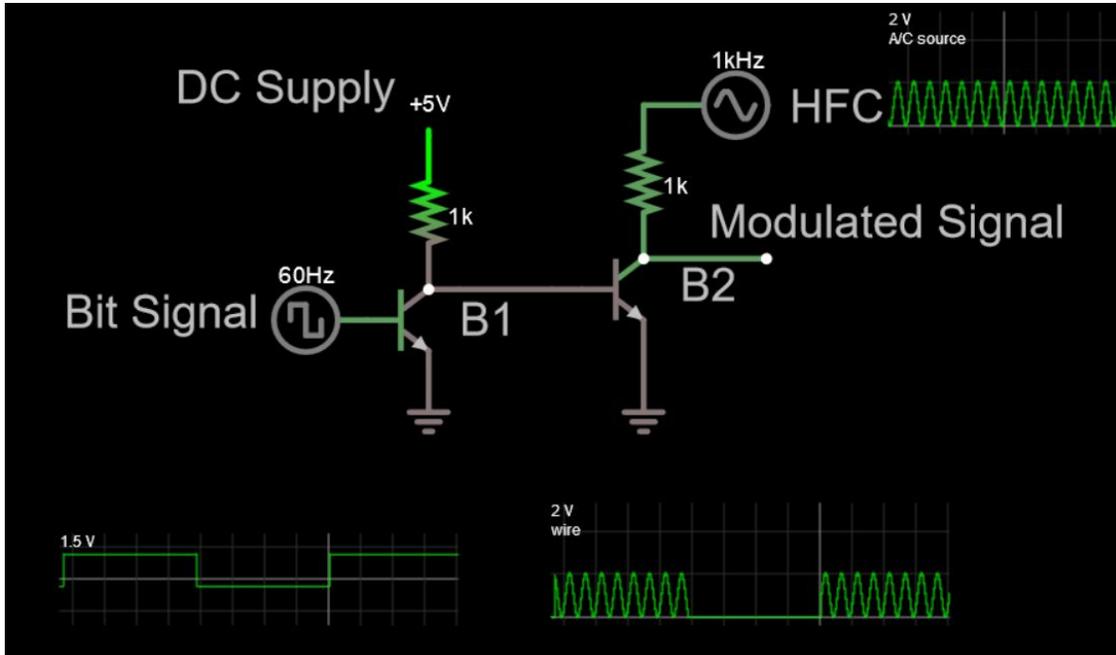


Figure 4. Falstad simulation of the BJT modulation circuit. The bit signal, HFC, and modulated signal outputs are also scoped.

2.2.2 Demodulator

The process of demodulation is modeled in Figure 5. Once injected and extracted, the modulated signal will have negative components. A rectifier is used to retain its strictly positive form. Then the signal is passed to a low-pass filter (LPF) to filter the HFC and get the envelope of the modulated signal. From there, MC2's built in comparator can read the demodulated signal now that it is digitized. For our LPF, we used the cut-off frequency equation is

$$f_c = \frac{1}{2\pi RC} \quad 2.2$$

where the R and C are the resistance and capacitance values of the LPF.

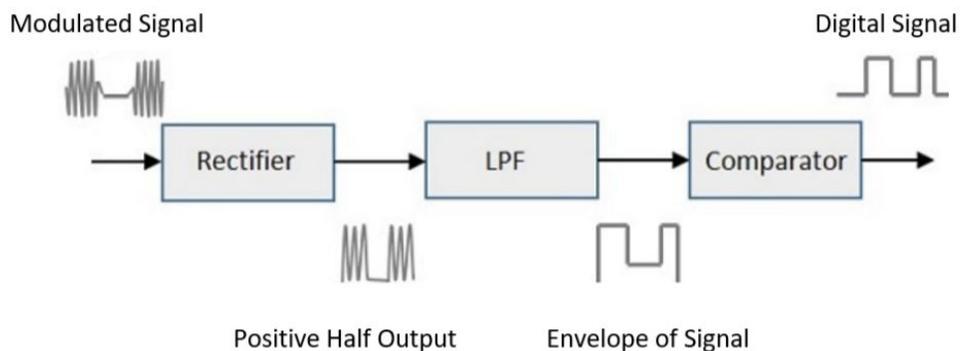


Figure 5. Demodulation Visualization

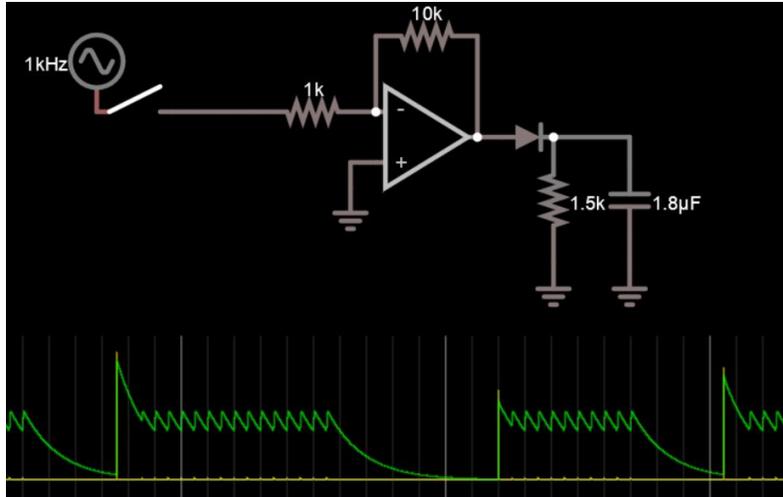


Figure 6. Falstad Simulation of Demodulation

In our simulation shown in Figure 6, we modeled the modulated signal as a switch controlled sinusoid to closer represent how our modulated data would look like instead of a 50% duty cycle signal. The output envelope is scoped below the circuit. We also used an inverting operational amplifier (opamp) to raise the voltage value of our modulated signal. Since Microcontrollers use a digital threshold to determine 1s or 0s by reading signals at or above the digital threshold, we needed to amplify our signal enough so its 1s were above the digital threshold and its 0s were below. The equation for a non-inverting opamp is

$$V_{out} = -V_{in} \left(\frac{R_f}{R_i} \right) \quad 2.3$$

Where R_f is resistor bridging negative opamp input and opamp output, and R_i is the input resistance between the relayed signal and the negative input. We realized after simulation that we needed non-inverting opamps. The equation for that configuration is

$$V_{out} = V_{in} \left(1 + \frac{R_f}{R_i} \right) \quad 2.4$$

The resistors here are placed different and will be shown in a later section. After the opamp, the envelope is mostly absent of high frequency components. The key features are the discharge time lengthening the logical 1s and the ripple voltage riding the tops of each logical 1. Discharge time is modeled by the equation

$$V(t) = V_o (e^{-t/RC}) \quad 2.5$$

Where $V(t)$ is the input of the capacitor, R and C are resistance and capacitor in series, and t is the discharge time.

$$V = V_{in} \left(1 - \frac{t}{RC} \right) \quad 2.5$$

Here, R and C is the resistor and capacitor of the LPF, V_{in} is the voltage input from the rectifier output and t is the total discharge time.

2.3 Master Transmitter Circuit Components

The master node required two 2N222A BJTs and a TLV2461 operational amplifier to properly modulate the signal. The non-inverting amplifier configuration is now seen in Figure 7. R_i is R18 on the schematic while R_f is R15. Note that for this resistor we use a 50M Ω potentiometer. The potentiometer allows us to adjust the filter. All other resistors were 1k Ω .

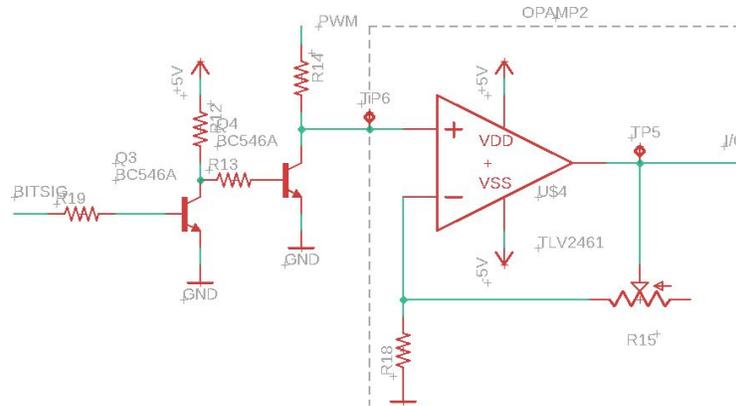


Figure 7. EagleCAD Schematic of the Modulator Circuit

The signals “BITSIG” and “PWM” (pulse width modulation) originate from a single ATmega328p microcontroller. BITSIG comes from a digital output pin while PWM also comes out of a digital PWM pin. One notable improvement would be the use of an operational amplifier with a higher voltage gain than the 6 V/V the current TLV2461 is limited to. The design still meets the verifications later discussed, but the higher gain range would positively impact the project since frequencies used are not high, allowing for more gain before exceeding an amplifier’s rated GBW (Gain-bandwidth product). We also noted that R13 between the base of one BJT and collector of the other is unnecessary. For our PCB, we replaced R13 with a wire. At test node TP5 is where we scoped our final modulated signal before power line injection.

2.4 Virtual Grid Components

To inject our modulated signal onto a 12V DC supply, we require a capacitor between the virtual grid powerline and the output of the modulator. This is to block the 12V DC from interfering with the Master node. We found a capacitor of 0.22 μ F sufficed. Another necessary component is the inductor leading to the 12V DC supply. The same way the capacitor blocks DC, the inductor acts as an impedance to our modulated signal. To find the value of impedance, we formed an equation that described the capacitance and inductance as impedances of a voltage divider.

$$\frac{V_{out}}{V_{in}} = \frac{j\omega L}{j\omega L + \frac{1}{j\omega C}}$$

ω is the frequency of our HFC, L is the inductance, and C is the capacitance 0.22 μ F mentioned above. The $V_{out}:V_{in}$ ratio would ideally be 1, since we want no loss in our modulated signal, but because we have

amplifiers in the Follower node, we allowed the ratio to be 0.1V/V so that our inductance value would not have to be too high. From this equation, we found an acceptable inductance value to be 824 μ H. For data extraction, we simply used another 0.22 μ F to again block the 12V DC supply. This capacitance is C3 of Figure 8.

2.5 Follower Node Circuit Components

After going through capacitors and inductors, the shape of our modulated signal changes and there is some amplitude loss. For this reason, we use another TLV2461 amplifier. For the rectifier we used an LN4001 diode. The LPF consisted of a potentiometer 50M Ω and a 0.47 μ F. Every potentiometer used in this project was 50M Ω and was used for adjustments. After the LPF, a second TLV2461 amplifier was used to amplify the envelope so it meets its digital threshold requirements. All opamps were adjusted with the potentiometer for maximum amplification, capping at 6V. We were mindful not to maximize the potentiometer, for clipping would occur and the signal would become distorted at its peak values.

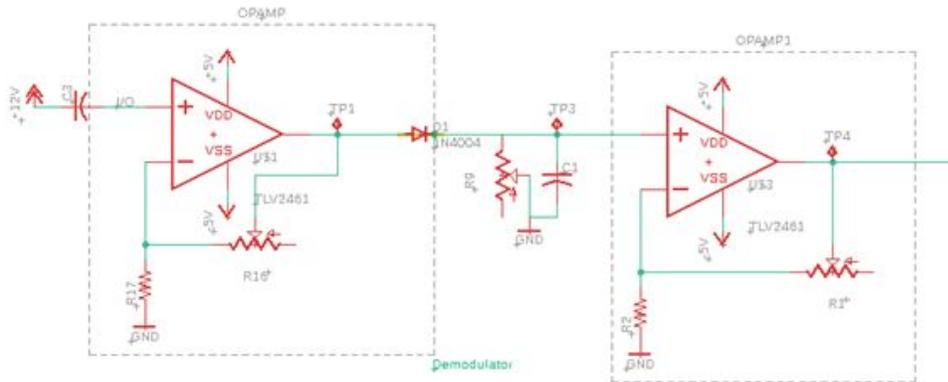


Figure 8. EagleCAD Schematic of Demodulation Circuitry

2.6 Microcontroller Code

The nodes each communicate via Universal Asynchronous Receiver/Transmitter (UART). Only the transmitting Tx pin of the master microcontroller needs to connect to the Rx receiving pin of the Follower microcontroller. An ATmega328p chip was chosen for its low cost, arduino compatibility and built in registers for multiple communication protocols for us to test out, including SPI and I2C. Empirically, each MC required a 16 Mhz crystal oscillator to function.

2.6.1 Microcontroller Master Scheduling Code

The master node sends its own schedules from an array buffer. The special beginning character is an ampersand (&). The subsequent characters in the array are data values which the Follower will actually read as the intended schedule data (Figure 9). For integration, this code outputs the “BITSIG” out of the modulator circuit of 2.2.

```
const byte redSchedule[8] = {'&', 0, 3, 0, 1, 0, 1, 0};

void setup() {
  Serial.begin(300); // opens serial port, sets data rate to 300 bps
```

```

}

void loop() {
  Serial.write(redSchedule, 8);
}

```

Figure 9. Code Block for the master microcontroller. Each element of the array “redSchedule” will be received one element at a time by the Follower until the full array is retrieved. When writing to multiple Followers, other schedule variables (e.g. “blueSchedule”) may be created.

2.6.2 Microcontroller Follower Code

The Follower continuously waits for input from its Rx pin. Once the Follower recognizes the special ‘&’ character, the subsequent data can either be used immediately by the Follower or be stored into another variable to set future schedules without requiring further communication from the master node. The signal into the follower corresponds to the I/O signal in the schematics of transmitter module 2.2 and receiver module 2.3.

```

int incomingByte = 0; // for incoming serial data
byte redSchedule[8];

void setup() {
  Serial.begin(300); // opens serial port, sets data rate to 300 bps
}

void loop() {
  if (Serial.available() > 0) {
    // read the incoming byte:
    incomingByte = Serial.read();

    if(incomingByte == '&'){
      Serial.print("I received this schedule: ");
      Serial.println(incomingByte);
      Serial.readBytes(redSchedule,4);
      Serial.println(redSchedule[0]);
      Serial.println(redSchedule[1]);
      // ... Can do similar things with indices 2-7
      // Store the values in a new global variable if desired
      Serial.println(redSchedule[8]);
      //Place differing control for digital output of LEDs here
      //digitalWrite(13, HIGH);
    }
  }
}

```

Figure 10. This code block for the Follower will correctly retrieve the schedule array sent by the master microcontroller. For this project, additional digital pins were set high in order to send a signal to a BJT to activate a relay.

2.6.3 Microcontroller PWM Code

This code was used to generate the high frequency carrier or “PWM” signal fed into the master transmitter circuit of 2.2.

```

#include <TimerHelpers.h>

```

```

const byte timer0OutputA = 6;
const byte timer0OutputB = 3; //pin 11 on ATmega

void setup() {
  pinMode(timer0OutputA, OUTPUT);
  pinMode(timer0OutputB, OUTPUT);
  TIMSK0 = 0; //no interrupts
  Timer0::setMode(5, Timer0::PRESCALE_1, Timer0::CLEAR_A_ON_COMPARE | Timer0::CLEAR_B_ON_COMPARE);
  OCR0A = 200; //number of counts for a cycle
  OCR0B = 100; //duty cycle within OCR0A, now 50%
}

```

Figure 10. This code uses a timer library [1] to set a 40 kHz frequency PWM and the on-chip “output compare” registers OCR0A and OCR0B to set a PWM duty cycle. Varying OCR0B from 0 to 200 will vary the duty cycle proportionally.

2.7 Relay and Appliances

A signal output from the Follower microcontroller enters the base of a BJT, which shorts the collector and emitter, and energizes the coil of a relay from normally closed to normally open, which allows a 12 V supply to connect to an appliance in this project. The circuit for this is shown in Figure 11.

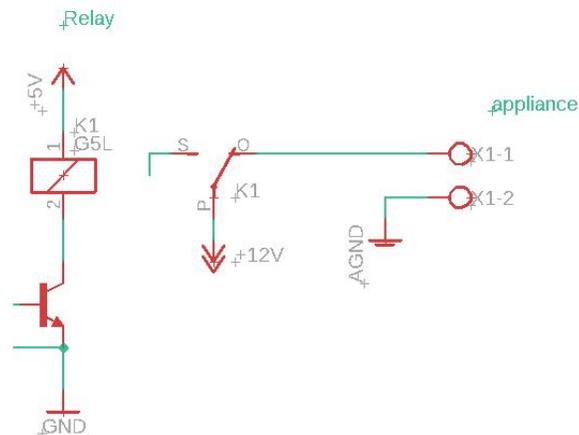


Figure 11. EagleCAD Relay Circuit.

In the actual project, we had a 12 V relay and the +5V signal needed to be a higher signal to energize the coil. The supply to node “P” could be a higher voltage than 12V as it represents the supply to a larger appliance which the microcontroller alone could not switch on or off with its lower voltage (< 5 V) output.

3. Signal Analysis and Verification

At each important step of our signal processing, the output was measured on an oscilloscope to verify requirements. The points were tested were at MC1 output, Modulation Circuitry output, the Follower node input, and finally Demodulation Circuit output

3.1 Master Node Verification

3.1.1 Microcontroller 1 Signal Generation Verification

MC1 was required to generate a data bit signal at 150Hz and a HFC of 40kHz. Figure 12 confirms that both signal frequencies are within 0.1% of their desired values

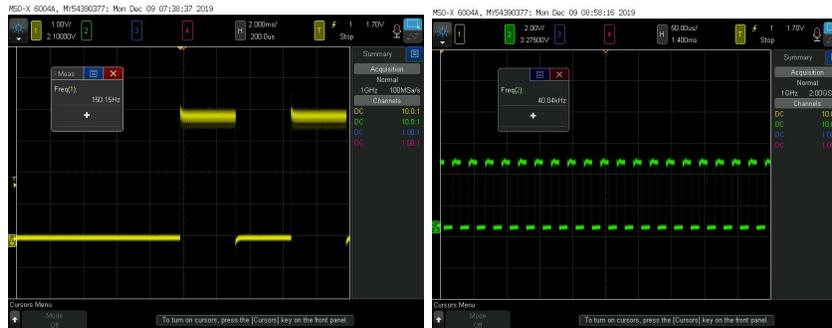


Figure 12. MC1 Bit Signal and HFC Generated by PWM

3.1.2 Modulation Verification

Proper modulation required that the modulation signal frequency be within 10% that of the original bit signal frequency. Tolerance is chosen to match UART's required 10% error tolerance for transmitted and received messages. Figure 13 displays the modulated signal



Figure 13. Modulated Signal

3.1.3 Powerline Injection and Data Extraction Verification

Scoping the input to the Follower node verifies that our signal was correctly injected and extracted. As we see in Figure 14, the signal represented our modulated signal except for the expected recentralization of 0, caused the signal to have negative components. We can also see the HFC is still present at 40kHz.

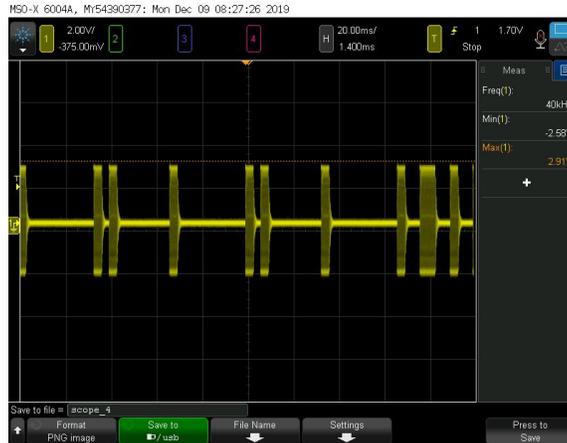


Figure 14. Follower Node Input

3.1.4 Demodulation Verification

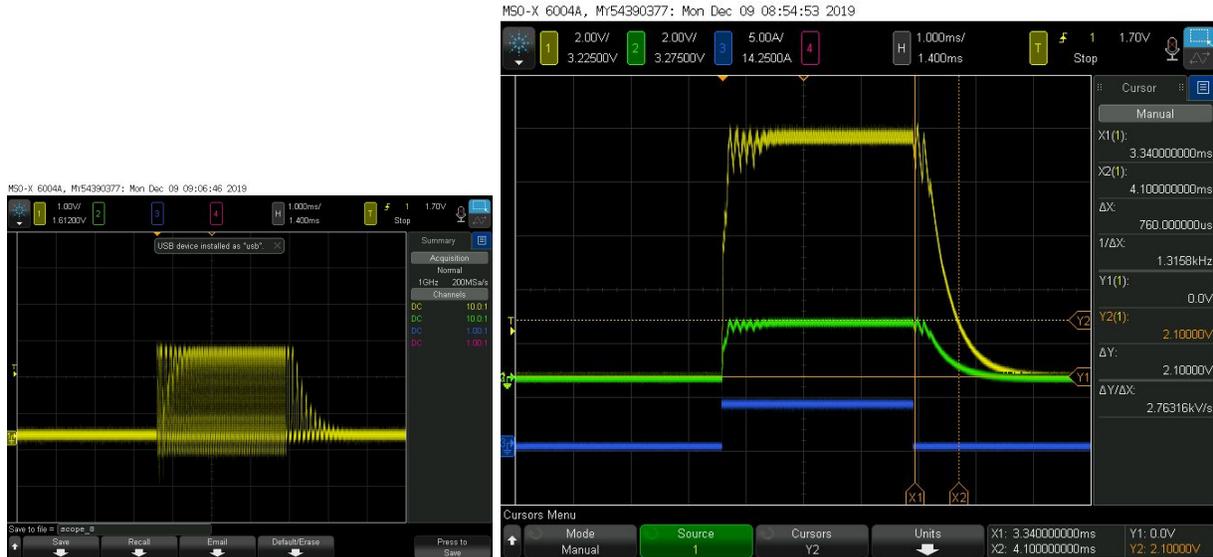


Figure 15. IRectifier Output and Comparison of Envelopes of Original Bit Signal

In Figure 15. we can see the effect of the rectifier on the left and the effect of the LPF on the right before and after amplification. Unfortunately, the digital 1 of our final envelope was measured to be 760 μ s longer than the original bit signal's digital 1 due to the capacitive discharge time. This 760 μ s was found by measuring the distance between initial discharge and the digital threshold for 0, at 2.1V (the digital threshold for the microcontroller powered at 4V). This is a 24% error difference and is the reason why our MC2 was not able to interpret the data correctly and thus not able to govern the relays on our PCB.

3.1.5 Relay Control Verification

As stated before, on our PCB, MC2 was not able to read the demodulated signal. However, if the data was instead directly sent on a wire between the Master and Follower, the Follower MC is able to interpret the data and control the relays. We used LEDs in series with current limiting resistors to

simulate appliances. We also had one LED that was connected directly to the simulated grid (not connected to the follower node) to demonstrate that unscheduled devices still worked properly.

3.1.6 Single Master Writer to Multiple Follower Reader Verification

Using the I2C protocol, a master node was able to send data schedules to multiple Follower microcontrollers (Figure 16) [2].

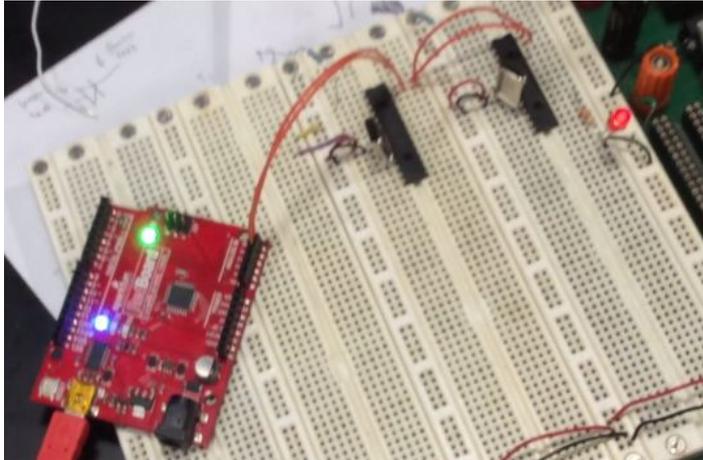


Figure 16. The center black ATmega328p chip is the master microcontroller. The other black chip on the right is a Follower, as is the RedBoard on the left. The lit up LEDs confirm the communication. Each Follower node has two orange cables connecting their input pins to the Master’s output. These correspond to the SCK and data signals for standard I2C.

4. Cost Analysis

4.1 Labor

Table 1. Cost of Labor

Engineer	Hourly Rate	Hours (Average of 40 hours/week x 12 weeks)	Hourly Rate x Hours x 2.5
Antonio Martinez	\$40.00	480	\$19,200
William Widjaja	\$40.00	480	\$19,200
			Total: \$38,400

4.2 Parts

Table 2. Costs of Labor

Description (hyperlinked)	Manufacturer	Part Number	Quantity	Unit Cost	Total Cost
------------------------------	--------------	-------------	----------	-----------	------------

Assorted Resistors, Capacitors, Wires (through hole)	Texas Instruments	Various	5 of each from 10, 20, 50, 100, 1k, 10k and 1M	\$0.40	\$14.00
1N4004 Diodes	Diodes Incorporated	1N4004DICT-ND	6	\$0.18	\$1.08
Arduino Uno Rev3	Mouser Electronics	8058333490090	2	\$22.00	\$44.00
Microcontroller	Microchip Technology	ATMEGA328-P U-ND	4	\$1.96	\$7.84
Operational Amplifier	Digikey	TLV2461 SOIC	10	\$1.74	\$17.40
12 V Relay	Digikey	Z1012-ND	4	\$1.33	\$5.32
2x3 Female Header	Digikey	S7071-ND	4	\$0.61	\$2.44
+5 Volt Regulator	Digikey	MC7805ACTG	4	\$0.60	\$2.40
-5 Volt Regulator	Digikey	30160500	4	\$0.60	\$2.40
16MHz Crystal	Digikey	COM-00536	4	\$0.57	\$2.28
2N222A BJT	Digikey	2N7000FS-ND	6	\$0.36	\$2.16
820 uH inductor	Mouser	80-SBCP-47HY821B	2	\$1.05	\$2.10
					Total: \$103.42

Table 2. Cost of Labor

Total Cost of Project: \$38,503.42

4.3 Schedule

Week of	Antonio	William
9/30	Work on Design Document, Simulate	Work on Design Document, Order Parts

	Circuits of modulator and demodulator	
10/7	Create and test breadboard circuit of modulator	Create and test breadboard circuit of bit signal generation
10/14	Create and test breadboard circuit of demodulator	Data generation circuit created and program written for sending bits between Arduinos.
10/21	Test the combined modulator, virtual grid and demodulator on a breadboard. Draft EagleCAD schematic.	Write PWM code and scheduling code. Breadboard and test the relay circuit (or simulate it if part has not arrived yet). Draft EagleCAD schematic.
10/28	Submit PCB order. Design and choose inductors and capacitors for virtual grid module.	Test PWM and scheduling with multiple nodes and “appliances” without the relay.
11/4	Solder PCB and test modulator and demodulator portions	Solder PCB and test microcontroller portions on Master and Follower
11/11	Integrate soldered PCBs and test the completed modulator and demodulator path at test points.	Fix any software issues or bad PCB connections. Solder on the relay and wires to connect to appliance LED
11/18	Prepare for Mock Demo and Mock Presentation. Solder the modulator and demodulator with virtual grid inductors and capacitors	Prepare for Mock Demo and Mock Presentation. Test code for multiple schedules and Follower nodes
11/25	Work on Final Presentation. Prepare for Demonstration by debugging connections or unexpected circuit/software behavior with William.	Work on Final Presentation. Prepare for Demonstration by debugging connections or unexpected circuit/software behavior with Tony.
12/2	Together: Take videos of each working component and rehearse the final presentation. Create poster.	Together: Take videos of each working component and rehearse the final presentation. Create poster.
12/9	Complete the final paper and teamwork evaluation	Complete the final paper and teamwork evaluation

5. Conclusion

5.1 Accomplishments

Our final implementation met expectations for each individual block and successfully transmits a signal with predictable error across a small imitation power line of long wires. Additionally, each module did work independently, including the relay control, schedule setting and Master writer \Rightarrow multiple Follower communication. The goal of sending a message across the power line was met.

5.2 Uncertainties

The interpretation of the demodulated data received at the Follower node is what caused issues when integrating the Follower module. As UART used within the ATmega328p generally requires the receiver and transmitter bauds to differ by no greater than 5%, a discharge time of the digital 1s exceeding 24% of the desired initial length would not allow for UART reception [3].

Since our design excluded transformers, it is empirically unclear how our signal would behave if it were to ride along high voltage step up and step down transformers, or even 1:1 isolation transformers. Most likely, we would need to study the core loss and skin effect caused by sending high frequencies through a transformer.

5.3 Ethical considerations

An obvious concern with this project is the electrical shock hazards and fire hazards caused by any connections with voltages above 5V. On such a small PCB which connects to high voltage devices, relying on trace widths alone would be insufficient to ensure reliable protection from unintentional shorts. One way of mitigating this risk include a feedback mechanism which activates a switch sending the high voltage supply to a different path if an unexpected voltage level is read at a point in the circuit.

We would also need to be careful about marketing to customers who cannot afford to live with scheduled power and warning any utility companies we work with to avoid marketing to these people by placing obvious disclaimers in their marketing materials and receiving written consent.

Data security is another concern. This could be mitigated by implementing public key cryptography where the Follower node has a private key and no other interceptors of data on the power line can get unencrypted data without the private key.

We would disclose all foreseeable concerns, following IEEE Code of Ethics #1 - "To hold the public safety first and to disclose factors of our project that might endanger the public" [4].

5.4 Future work

Our project has the potential to be quite beneficial for utility companies seeking better efficiency and customers trying to save money. Our issue with demodulating the data to look more like the initial data can be solved through the use of more ideal components such as higher gain amplifiers and a full-wave rectifier with low threshold voltage diodes. These would reduce the effect of amplitude losses. Two

methods of improving the error rate caused by the capacitive discharge include adding a higher order low pass filter to more sharply cut off the fall time of the digital 1s or to simply reduce the frequency of the data signal. The relay could also be a 3 to 4 V relay instead of the 12 V rated one for this project. This would allow a lower voltage to energize the coils.

Once the powerline communication design has been perfected, we could then focus on plugging the Master and Follower nodes to the outlets of homes. At that point, our project would be complete at least in how we initially envision it. More could be implemented into powerline communication such as schedule data encryption and two way communication. This would allow the party responsible for schedules to get a better insight into power usage of individual homes or neighborhoods.

References

- [1] Timers and counters - Gammon Forum. [Online]. Available: <http://www.gammon.com.au/timers> [Accessed: 19-Nov-2019].
- [2] Master Writer/Slave Receiver. [Online]. Available: <https://www.arduino.cc/en/Tutorial/MasterWriter>. [Accessed: 17-Nov-2019].
- [3] Atmel, "ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash, Table 19-2. Recommended Maximum Receiver Baud Rate Error for Normal Speed Mode", [Online]. Available: <https://www.microchip.com/wwwproducts/en/ATmega328p>. [Accessed: 13-Oct-2019]
- [4] IEEE, "IEEE Code of Ethics," 2019. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>. [Accessed: 8-Dec-2019]

Appendix A Requirement and Verification Table

Power Supply

Requirements	Verification
1. Components requiring power to function are appropriately powered and working.	1. LED indicating microcontrollers are on are lit and measured output of OpAmp is amplified by desired value

Microcontroller 1 (Master)

Requirements	Verification
1. Generates bit signal at frequency much lower than oscillating signal (i.e. 60Hz : 1kHz)	1. Program MC1 to output bit signal so we can display and measure its frequency using an oscilloscope while also confirming the correct sequence of bit signal.
2. Generates carrier oscillating signal at high frequency (> 500Hz)	2. Program MC1 to output oscillating signal using PWM and measure it with an oscilloscope.
3. Changes bit signal based on seasonal demand	3. Verify change in bit signal after an intended period using oscilloscope

Modulator Circuit

Requirements	Verification
1. Modulates bit signal with on chip oscillating signal and outputs modulated signal with correct frequency (i.e. 60Hz for 60Hz bit signal)	1. Record output signal on oscilloscope and confirm its frequency while also verifying correct bit shape of test signal from waveform generator.

Operational Amplifiers

Requirements	Verification
1. OpAmp successfully amplifies by desired amount	1. Use oscilloscope to measure input and output of OpAmp.

Virtual Grid

Requirements	Verification
<ol style="list-style-type: none">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.2. Modulated signal is successfully transmitted by master side on supply line and is read by slave side.	<ol style="list-style-type: none">1. Measure signals at each electrical node and ensure there are no frequency components above 500 kHz2. Measure modulated signal on scope before and after going on to the supply line and verify the overall shape of the signal retained.

Demodulator Circuit

Requirements	Verification
<ol style="list-style-type: none"> 1. Must correctly attain the envelope of the modulated signal 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. 	<ol style="list-style-type: none"> 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.

Microcontroller 2 (Slave)

Requirements	Verification
<ol style="list-style-type: none"> 1. Must store correct bit representations as intended by transmitter/receiver and properly control the relay switch with those signals. 	<ol style="list-style-type: none"> 1. Slave accepts input schedule data. Use serial terminal to print this input. 2. Scope output pin or place LED/device at input to the relay in case relay module fails. Verify a schedule of power is being given that matches the code.

Relay

Requirements	Verification
<ol style="list-style-type: none"> 1. Must open and close a circuit with no more than 5s delay. 2. Should properly dictate the power of connected appliances 	<ol style="list-style-type: none"> 1. Send a test input and measure the time it takes for the relay to react 2. LEDs demonstrate that they follow their intended schedules