

Exercise Bike Generator with Grid Synchronization

ECE 445 Final Report

Team 3:

Elisa Krause (elisak2)

Raihana Hossain (rhossa2)

Tiffany Wang (tw22)

TA: Bonhyun Ku

Fall 2021

Abstract

This final report describes the components of our grid-synchronized exercise bike generator and our design process. We start off with introducing our motivation for this project and the main components of our bike generator. We then discuss the high-level requirements that would deem our project to be successful. In addition, we describe the sub-requirements that were meant to help us achieve the high-level requirements, and we bring up which ones we were not able to successfully complete and why. Following that, we go over the costs that are relevant to this project. At the end, we conclude our report with what we would change or add if we were to continue this project.

Table of Contents

1. Introduction	1
1.1. Purpose	1
1.2. Visual Aids	1
1.3. High-Level Requirements	2
1.4. Subsystem Overview	3
1.4.1. Block Diagram	3
1.4.2. External Interface Devices	4
1.4.3. Generator	4
1.4.4. Control & Metering	4
1.4.5. Grid Safety	4
2. Design	5
2.1. Generator Module	5
2.2. Control & Metering Module	8
2.3. Grid Safety Module	9
3. Cost & Schedule	11
4. Design Verification	12
4.1. DC-DC Converter	12
4.2. Inverter	13
4.3. Relay	14
4.4. Sensors	14
4.5. Display	15
5. Conclusion	17
5.1. Accomplishments	17
5.2. Uncertainties	17
5.3. Future Work / Alternatives	18
5.4. Ethical Considerations	18
References	19
Appendix A: Overall Schematic and Display Code Flowchart	21
Appendix B: Parts & Costs Table and Schedule	24
Appendix C: Requirements & Verification Tables	26

1. Introduction

1.1. Purpose

Something we take for granted everyday is energy. There is energy consumed constantly in malls, offices, schools, vehicles, gyms, and more. A special feature about gyms is that there is almost always someone using the exercise machines like the elliptical, treadmill or bike. Now what if, along with losing those extra pounds, the participant can also generate some electricity using these machines? Our device is a straightforward and cheap alternative for gyms to have retrofitted spin bikes that generate electricity, and for the gym to save money by connecting the exercise bike to the gym's grid by simply plugging the device into the wall outlet.

Bike generators have been made in the past, and they typically produce energy for a very specific and limited function like charging a battery, lighting up LEDs, etc. However, the difference with our project is that we are sending the energy to the grid. This is something that will expand the range of what the generated electricity from the bike can be used for.

1.2. Visual Aids



Figure 1 (left): Box containing PCB and metering components

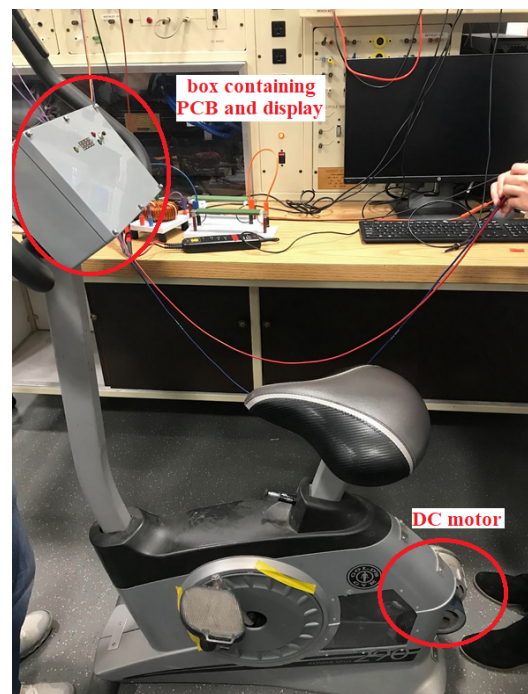


Figure 2 (right): Bike with all physical components of project

1.3. High-Level Requirements

- **[Electricity Generating Efficiency]** Rider is able to generate at least ~~40 W~~ 10 W of electricity while pedaling exercise bike at moderate biking speed
- **[Grid Synchronization]** Power output from bike is able to synchronize to the gym's grid
- **[Metering Success]** Display is able to show current power being generated and total energy generated since the last reset

[Electricity Generating Efficiency]

The average biker can generate around 100 W for about an hour on a bike [1]. Due to inefficiencies from harvesting this power, not all of the energy can be converted into electricity. The power consumption of a TV is usually 80-400 W, and a typical 42-inch LED TV consumes about 80 W [2]. Based on these numbers, we want to verify that our efficiency is good enough by having one of our group members use the spin bike at a moderate speed for 1 minute and maintain a power output of at least 40 W, which is half of the required power needed to run a low-power TV. This success can be verified with our display.

The target 40W has been modified to 10 W per the suggestions of professors and TAs due to the possibility of safety being compromised from aiming for 40 W.

[Grid Synchronization]

The final bike output signal would be synchronized with the grid. That is, the frequency, voltage and phase of the signal would line up with the grid. An AC voltage generator would be used as a reference for our final bike output to match (we will not actually be plugging our device to the grid).

Our bike would be considered a distributed energy resource (DER). IEEE 1547-2018 [3] is the IEEE standard for interconnections between DERs and the grid. Table 1 shown below from IEEE 1547-2018 shows the maximum difference in frequency, voltage and phase angle between our output and the grid. Our system would be rated less than 500 kVA, so the first row of the table below provides the corresponding requirements.

**Synchronization parameter limits for synchronous interconnection to an EPS,
or an energized Local EPS to an energized Area EPS**

Aggregate rating of DER units (kVA)	Frequency difference (Δf , Hz)	Voltage difference (ΔV , %)	Phase angle difference ($\Delta \Phi$, °)
0-500	0.3	10	20
> 500-1 500	0.2	5	15
> 1 500	0.1	3	10

Table 1: Bike Output Requirements

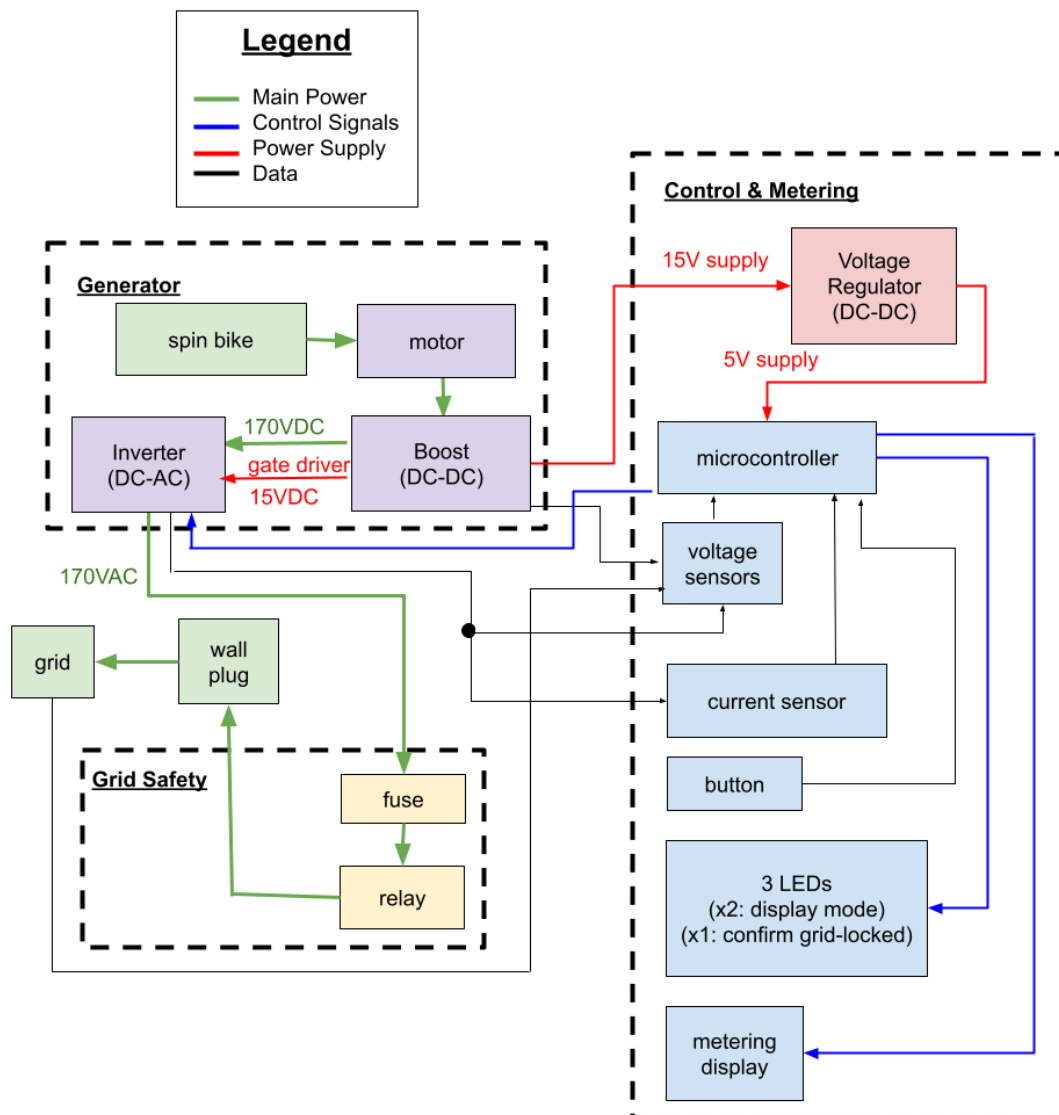
[Metering Success]

This is based on our electricity-generating success in the first high-level requirement (Electricity Generating Efficiency). The display in Mode 1 would show the current power output while someone is biking, and after pressing the button which switches the display to Mode 2, the display would read at least 0.167 Wh after 1 minute of biking. In addition, the total energy generated would reset to 0 when the button is held down for 2 seconds.

1.4. Subsystem Overview

There are two main components of our bike generator. The first main component converts our bike motor's voltage into a form that allows the power generated by the bike to be placed onto the grid. That is, the motor voltage is converted into 120 VAC 60 Hz and is also synchronized with the grid's voltage. The second main component is a user interface that allows bikers to see how much power and energy they are putting onto the grid. Both of these components involve interaction between the “Generator” module and the “Control & Metering” module, as it can be seen in Section 1.4.1. below.

1.4.1. Block Diagram



1.4.2. External Interface Devices

Grid (Simulated): The bike generator sends power to the gym's grid. For testing, the bike did not actually connect to the power grid. Instead, the "grid" signal was generated using a function generator, a sinusoidal signal of 60 Hz. Our device was attached to this signal to sync up with it.

Wall Plug: This is the plug that connects the grid-synchronized power generated by the bike to the grid using the wall outlet.

1.4.3. Generator

Spin Bike and Motor: The motor attaches to the wheel of the spin bike. When the spin bike is pedaled, the motor outputs a DC voltage that varies based on how hard the bike is being pedaled.

Boost Converters: One boost converter converts the varying DC voltage from the motor into a constant DC voltage with its magnitude corresponding to the peak of the final output sinusoidal signal. The second boost converter converts the varying DC voltage from the motor into 15 VDC, which is used for the 5 VDC voltage regulator and also to power the gate drivers.

Inverter: The inverter converts the constant DC voltage from the first boost converter into an AC signal. This signal lines up with the grid's signal (frequency, phase, amplitude) when synchronized and so the power can be sent to the grid.

1.4.4. Control & Metering

Voltage Regulator: This converts the 15 VDC power supply to a 5 VDC. The 5 VDC is needed to power the IC chips and other components on the board that require a 5 VDC power supply.

Microcontroller: This takes in the voltage and current information, processing it so that the appropriate inverter control signals and display control signals are generated.

Voltage and Current Sensors: These sensors convert the grid voltage, inverter voltage, and inverter current into appropriate voltage ranges that can be processed by the microcontroller.

Display System (Button, LEDs, HEX display): The display system is the user interface for the biker to see how much power they are currently generating, how much energy they have generated, and if the inverter output is synchronizing with the grid.

1.4.5. Grid Safety

Fuse: The fuse will supply feedback protection in case a dangerous amount of power is received from the grid.

Relay: If there is no power being generated, the relay will disconnect our device from the grid.

2. Design

See Appendix A for overall schematic.

2.1. Generator Module

Boost Converter

The boost converter takes the varying DC voltage from the DC motor and creates a constant 170 VDC. The input range of the DC motor is roughly [10 V, 40 V]. Due to the large difference between the lowest input voltage and the desired output voltage, a flyback topology was chosen for the converter. The turns ratio of the transformer in a flyback topology can be used to increase the output voltage of the converter. For a flyback converter, the relationship between the input voltage (V_{in}) and output voltage (V_{out}) is

$$\frac{V_{out}}{V_{in}} = \frac{D}{1-D} * \frac{N1}{N2} \quad (1)$$

where D is the duty cycle and N1/N2 is the turns ratio between the primary and secondary windings of the transformer.

Finding a transformer was somewhat difficult. The transformer should have as large a magnetizing inductance as possible to minimize current ripple. This calls for a gapped core [5]. Many of the transformers with appropriate turns ratios were meant for ac-ac conversion, meaning they had ungapped cores. It was briefly considered to run the inverter circuit at a lower voltage and include a transformer at the end to step up the voltage. However, this was decided against because the current in the circuit would be larger. By running the inverter at a higher voltage, the high current is isolated to the primary side of the flyback converter. A transformer with a turns ratio of $\frac{N2}{N1} = 8.62$ was chosen.

Another advantage of the flyback topology is the opportunity to have multiple outputs. We chose another transformer to place in parallel with our main transformer to produce a secondary output. The turns ratio of the second transformer was chosen to be $\frac{N2}{N1} = 0.5$ to create an output of 15 V. This voltage was designed to power the 5 V voltage regulator for the microcontroller, as well as power the gate drivers for the inverter.

Initially, our design included an RC snubber. The values for the snubber were calculated to reduce the voltage spikes [6]. The voltage across the snubber was equal to

$$V_{snub} = \frac{N1}{N2} * (V_{out} + V_f) \quad (2)$$

where V_f is the forward voltage of the diode. We chose this value to be equal to 20V. Then the R value was calculated:

$$R_{snub} = \frac{V_{snub}^2}{1/2 * L_{lk} * (I_{max,pri})^2 * \frac{V_{snub}}{V_{snub} - N1/N2 * (V_{out} + V_f)} * f_{sw}} \quad (3)$$

where L_{lk} is the leakage inductance of the transformer, $I_{max,pri}$ is the maximum current through the primary side of the transformer and f_{sw} is the switching frequency. The value of C could then be calculated as:

$$C_{snub} = \frac{V_{snub}}{\Delta V_{snub} * V_{snub} * R_{snub} * f_{sw}} \quad (4)$$

where ΔV_{snub} is the ripple voltage across the snubber. R and C were calculated to be 20 Ω and 470 nF respectively. During verification, we found the snubber circuit was limiting the voltage too much, preventing the correct operation. Additionally, the resistor and capacitors did not have a high enough power rating for safe operation, so we removed this portion in our final design.

The converter was controlled by LM5156, an analog boost controller. A reference design was provided in the LM5156 datasheet, and the values of the necessary resistors and capacitors were calculated using the LM5155/56 Flyback Controller Quick Start Calculator provided by Texas Instruments [7]. A few key features of the LM5156 made it ideal for this application. Aside from its wide input voltage range, it is able to internally generate the supply voltage for its gate driver from the input voltage. This means at low voltages when switching needs to start but the auxiliary output of the boost converter is still at 0V, the IC can use the input voltage to drive the MOSFET. Additionally, the LM5156 has a soft start which will help limit the transformer inrush current and protect the circuit.

Inverter

The inverter takes the DC voltage output from the boost converter and converts it into a sinusoidal AC signal with a peak value of 170 V. Two circuit topologies were considered for the inverter: a full bridge and a half bridge. A half bridge circuit would require the DC input voltage to be boosted up to 340 V to avoid a peak voltage of 170 V and an RMS voltage of 120 V at the output. To avoid this high voltage, we chose to use a full bridge inverter.

The MOSFETs were sized for 2 times their stress values. Consequently, we determined the minimum voltage rating to be 340 V and the minimum current rating to be 1.75 A. Two gate drivers controlled the MOSFETs. A bootstrap capacitor was required for each of the gate drivers. The value of this bootstrap capacitor was calculated using the gate charge of the MOSFETs and the gate voltage. The minimum value of the bootstrap capacitor needs to be ten times larger than the gate capacitance [8]. The gate capacitance was calculated by:

$$C = \frac{Q_g}{V_g} = \frac{24 \text{ nC}}{10 \text{ V}} = 2.4 \text{ nF} \quad (5)$$

The capacitance of the bootstrap capacitors was chosen to 33 nF $\pm 10\%$.

The microcontroller generated the two control signals for the gate drivers, q1 and q3. Initially, we designed q1 to be a sinusoidal signal while q3 was a square wave that ensured the output of the inverter was both positive and negative. However, after further research, we found we could achieve a less distorted output by also varying q3 sinusoidally, 180° out of phase with q1. The simulation results shown below in Figure 3 demonstrate the advantage of the sinusoidal q3 signal.

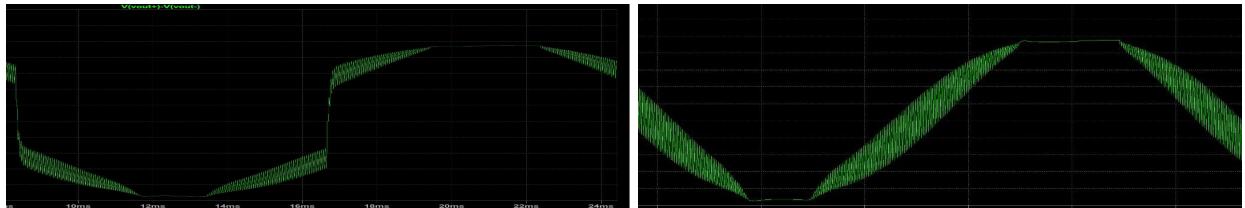


Figure 3: Output of inverter with square wave q3 (left) and sinusoidal q3 (right)

The open loop transfer function can be written as

where K_d is the gain of the phase detector, K_v is the gain of the lead compensator, z is the zero of the lead compensator and p is the pole of the lead compensator. The crossover frequency of the open loop transfer function was chosen to be at least 5 times smaller than the operating frequency of 60 Hz, or 377 rad/s. After adjusting z , p , and K_v , the final crossover frequency was designed to be 64 rad/s with a phase margin of 37° . A phase margin of 37° was good enough to ensure stability of the system even with nonidealities in the model. Additionally, a higher crossover frequency corresponds to a faster response which is necessary to minimize the phase difference between the grid voltage and the output voltage.

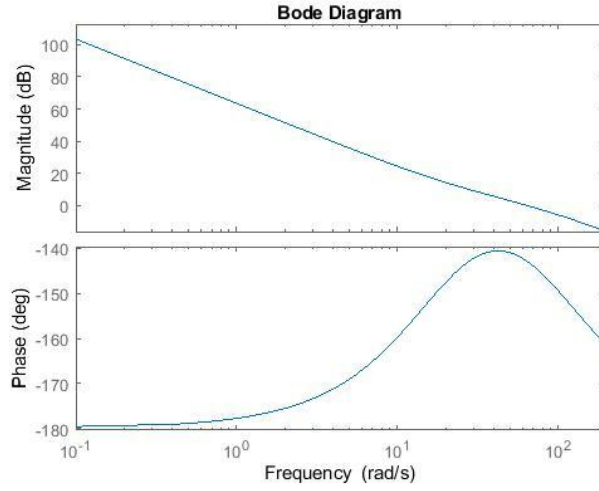


Figure 5: PLL bode plot

2.2. Control & Metering Module

Sensors

The voltage sensor circuits convert the inverter AC voltage, grid AC voltage, and the DC boost's voltage from $[-170 \text{ V}, 170 \text{ V}]$ or $[0 \text{ V}, 170 \text{ V}]$ to $[0 \text{ V}, 5 \text{ V}]$, since the microcontroller can only process voltages in the range of $[0 \text{ V}, 5.5 \text{ V}]$ according to its datasheet [4]. Figure 6 shows how the inverter voltage is converted through the sensing system to be able to be processed by the microcontroller.

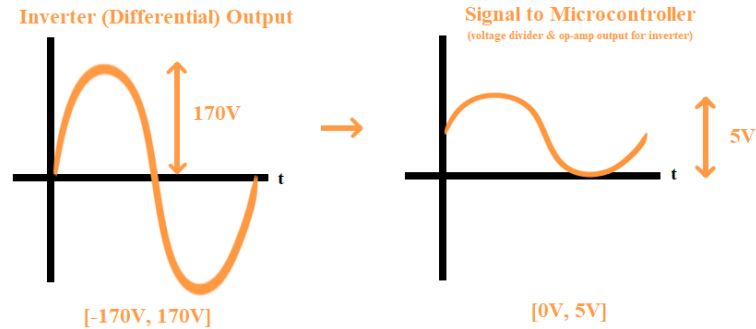


Figure 6: Scaling and shifting factor for initial design of inverter voltage sensing circuit

The voltage sensor circuits consist of voltage dividers and op-amps used to scale and shift the signals. The level-shifting circuit is based on an online calculator [9]. The differential op-amp design is based on the information in this website [10].

There is also a differential op-amp for the inverter since it has two output voltages (as seen in Figure 7), and the microcontroller cannot measure differential voltages.

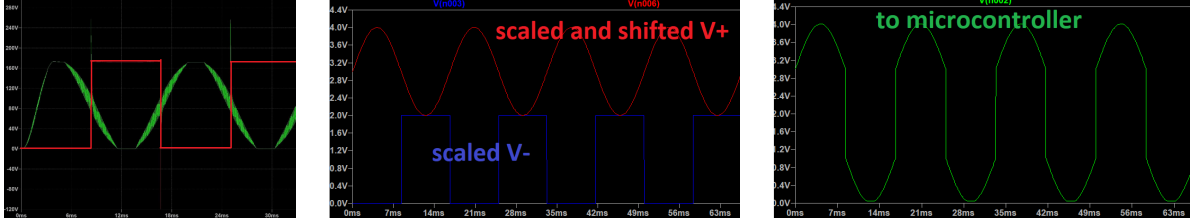


Figure 7 (left): Inverter's differential signals $V+$ (green sinusoidal wave) plotted with $V-$ (red square wave) resulting in 170 VAC

Figure 8 (middle): Simulation results of scaled and shifted $V+$ and $V-$

Figure 9 (right): Differential of the scaled and shifted $V+$ and $V-$ signal

The voltage dividers and op-amps were initially designed to convert $[-170 \text{ V}, 170 \text{ V}]$ or $[0 \text{ V}, 170 \text{ V}]$ to approximately $[0 \text{ V}, 5 \text{ V}]$, but because we ended up lowering our output voltage so that our inverter has a 19 V peak output, our readings were heavily affected by noise. As a design alternative, we increased the resolution of the sensing circuits by designing them for 50 V.

The software consists of two calibration values for each sensor circuit and converts the ADC value interpreted by the microcontroller into the original voltage or current.

Display, Button, LEDs

The design for the display system mainly involved switching between modes, calculating power and energy, and displaying numbers onto the HEX display. Switching between the modes using the button involves setting the initial button state to Mode 1 and a statement that sets the button to Mode 2 if originally in Mode 1 and vice versa. Mode 1 displays the current power. This is calculated by simply multiplying the current and voltage values calculated from the sensing code. Mode 2 displays the total energy. In order to calculate this value we found the average power and multiplied that by the time duration. See Appendix A for flowchart that summarizes the display code.

The SevSeg library that was used for our 4-digit 7-segment HEX display needed to be modified because some of the ATmega32U4's pins could not be accessed with shortcuts, and so the pinMode and digitalRead functions could not be used. Since the SevSeg library used these functions, we had to modify the SevSeg library and replace the pinMode and digitalRead functions with new functions in the .cpp file that assigned appropriate values to DDRx and PORTx.

2.3. Grid Safety Module

Relay

We used a relay with a 5 VDC coil voltage and 40 mA coil current.

The microcontroller has a limited amount of current that it can draw per pin, so this current needs to be amplified with a BJT in order for the relay to close. Based on the microcontroller datasheet [4], its absolute maximum current per pin is 40 mA, but also, the sum of a combination of pins in specific ports cannot exceed 100 mA. We had assigned the relay

output control signal to be a pin on Port B, and that pin contributes to the current sum with 9 other pins. Assuming that all those pins have an equal amount of current draw, this means that one pin cannot exceed $100 \text{ mA} / 10 \text{ pins} = 10 \text{ mA}$. To be on the safe side, we decided to have the pin draw 4 mA. The resistances for the BJT current-amplification circuit were determined based on this 10 mA current being amplified to 40 mA (relay coil current). The relay was modeled in LTSpice using [11] as a reference, and the simulation results are shown in Figure 10 below.

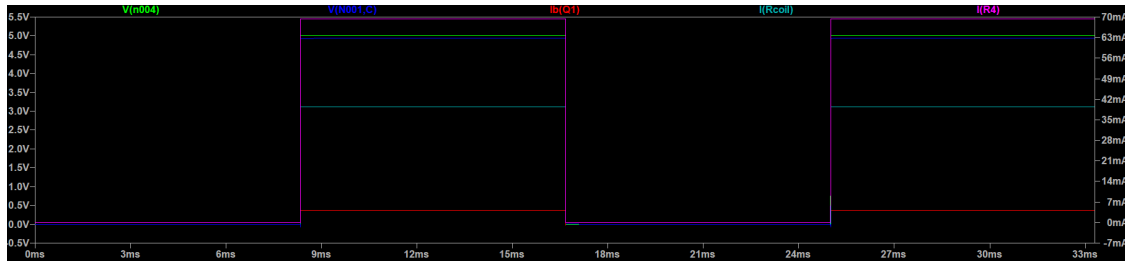


Figure 10: The relevant signals are the red, teal, and magenta signals

Red: Current coming from the microcontroller

Teal: Amplified microcontroller current that turns on/off the relay

Magenta: Shows a non-zero current when the relay closes

3. Cost & Schedule

Costs

Parts cost table in Appendix A

Overall parts cost (based on parts cost table) = **\$120.52**

Machine Shop

This project will take them about **20 hours**.

According to UIUC's machine shop website, the average cost is **\$36.65/hr plus materials** [12]

Total machine shop cost is $20 \text{ hr} \times \$36.65/\text{hr} = \733

Labor

Average salary of ECE graduate is **\$79,714/year** [13]

Average person works **2080 hours for full-time per year**

$$\$79714/2080 = \$38.32/\text{hr}$$

$$\$38.32/\text{hr} \times 2.5 \times 90 \text{ hr} = \$8,622 \text{ per person}$$

$$\text{Total Labor cost: } \$8,622 \times 3 = \$25,866$$

Total Project Cost

$$\text{Parts} + \text{Machine Shop} + \text{Labor} = \$120.52 + \$733 + \$25,866$$

\$26,719.52

Schedule

See Schedule in Appendix A

4. Design Verification

The Requirements & Verification Tables are in Appendix C.

4.1. DC-DC Converter

-DC-DC boost converter must be able to output $\pm 70\text{VDC} \pm 5\%$ **19 VDC $\pm 5\%$** .

We were advised to decrease our voltages for safety reasons, hence why we changed our requirement to 19 VDC. We were able to achieve 19 VDC as the entire input voltage range, as it can be seen in Figure 11 below.

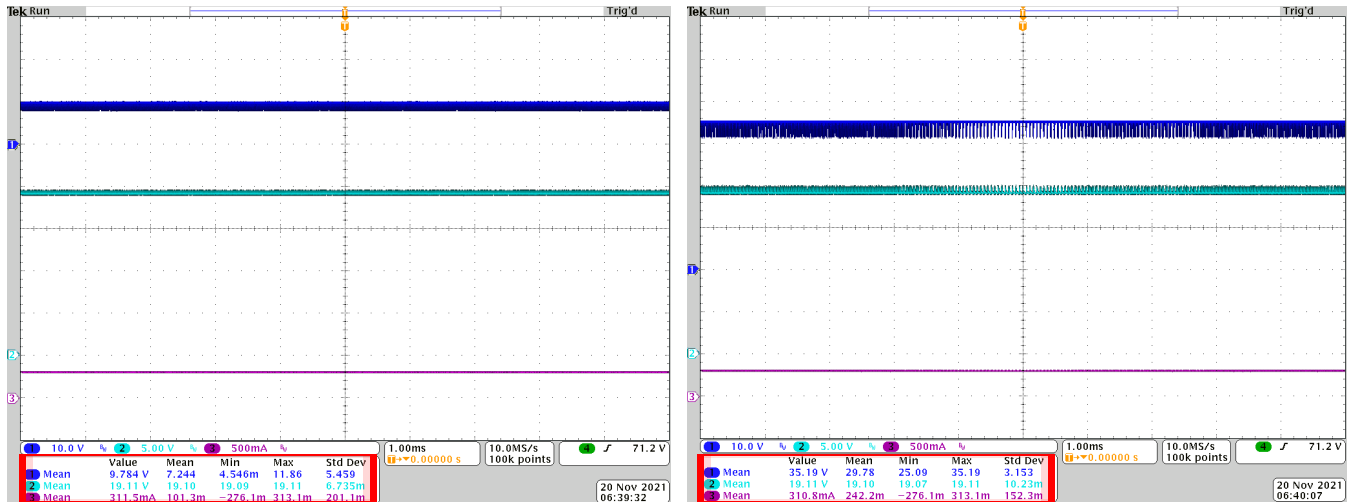


Figure 11: Boost converter output with an input of 9.8 V (left) and an input of 35.2 V

Blue: Input voltage

Teal: Output voltage

Purple: Output current

-DC-DC converter must be able to generate a $15\text{ V} \pm 5\text{ V}$ supply for the voltage regulators.

Since we reduced our primary voltage output to 19 VDC, the turns ratio between the primary and secondary output was incorrect. The secondary output was too low to power the 5 V voltage regulator and the gate drivers. However, even if the turns ratio was correct, we are uncertain the secondary output would have been successful. There was a lot of ringing in our boost converter circuit; ideally the voltage across the switch, shown in Figure ****, should be a square wave. This ringing was due to parasitics arising from long traces on the PCB as well as from our transformer not fitting into our PCB due to incorrect datasheet measurements. The ringing had very large peaks, and for the safety of our circuit, even if the turns ratio was correct, we decided not to implement the 15 VDC source and used an external voltage generator to power that source. We determined the ringing was from long traces when we were first testing the boost converter. Initially, the transformer was separate from the PCB and we connected it using wires. This way we could monitor the transformer current during operation. When we soldered the transformer onto the PCB, the ringing was reduced. Additionally, after we decided not to

implement the secondary output and removed the second transformer from the PCB, the ringing improved again. It is possible that a resonant circuit was formed with the two leakage inductances of the transformers, leading to more oscillations.

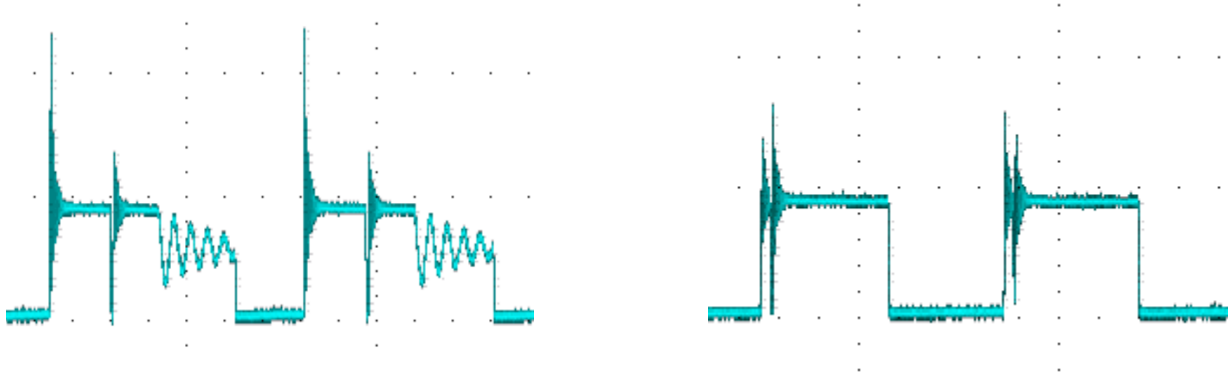


Figure 12: MOSFET V_{ds} ringing with 20V divisions before (left) and after (right) second transformer was removed

4.2. Inverter

-Inverter must be able to output $19/\sqrt{2}$ Vrms $\pm 10\%$ with a power factor of 0.7 or greater.

As it can be seen in Figure 13 below, our power factor is approximately 0.93, which satisfies the first requirement.

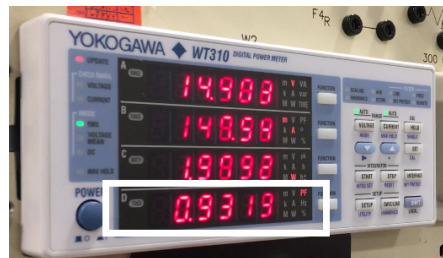


Figure 13: Power meter showing the PF of the inverter output

The final output of our inverter can be seen in Figure 14. The peaks of the output voltage are $\pm 19V$ are desired. The output voltage is more distorted than the output current, but this is expected as there was no output capacitor to limit the voltage ripple.

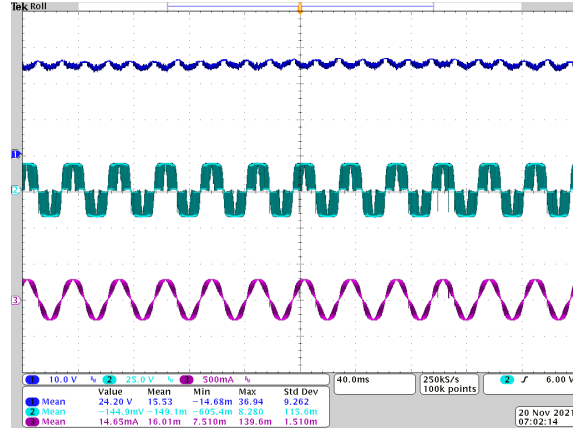


Figure 14: Input and output waveforms of the inverter

Blue: Input voltage

Teal: Output voltage

Purple: Output current

-Output voltage frequency must be able to synchronize to an external sinusoidal signal's frequency within $\pm 10\%$ and have no more than a 20° phase angle between the two.

This requirement was achieved using the Arduino Uno, but the PLL grid-synchronization algorithm was not integrated into our final product due to limitations of our microcontroller. That is why this requirement is considered to be partially achieved. The frequency and phase of the averaged inverter control switching signals– which corresponds with the inverter output– can be seen below to have achieved the frequency and phase requirement.

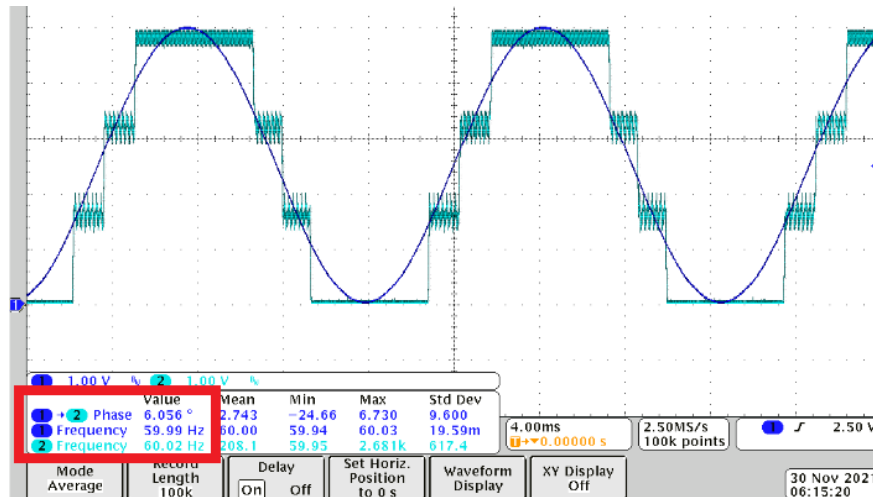


Figure 15: PLL algorithm results with the Arduino Uno

Teal: Averaged inverter control switching signals

Blue: Grid reference signal

4.3. Relay

-The relay disconnects the grid from our device within 2 seconds of inactivity of the bike.

This actually did not need a conditional statement in the code for this requirement to be fulfilled. Since the 5 V voltage regulator was no longer being powered when the bike stopped generating a voltage, then all the components on the board were no longer being powered, including the relay, which opened its contacts.

4.4. Sensors

-Grid voltage measurements processed by the microcontroller must have an accuracy of $\pm 5\%$.

Grid Voltage Input	Grid Voltage Read
0V	0.11V
10V	10.11V
20V	20.12V

Table 2: Grid voltage input and voltage processed by microcontroller

Initially, both the grid and the inverter sensing circuit had to be re-calibrated every couple of minutes since the sensing circuit was initially designed for 170 V and it was difficult to read the small voltages that we ended up using (19 V), due to the noise and also the op-amp outputs being sensitive to temperature changes. After scaling the sensing circuit down to be 50 V and thus increasing the resolution of the system, the grid voltage only had to be re-calibrated every couple days and was able to satisfy the first requirement when calibrated.

-Inverter voltage measurements processed by the microcontroller must have an accuracy of $\pm 5\%$.

On the other hand, due to the inverter's long path length from its output to the sensing system and also due to having more components involved with the inverter's output compared to the grid, there was a lot of noise for the inverter sensing circuit. Due to this noise along with the temperature sensitivity of the op-amps' outputs, this second requirement was not fulfilled. When the PCB was placed into our enclosure box, the temperature of the circuit rose higher and quicker due to poor ventilation, and the inverter sensing circuits only worked once a day for a couple minutes every time the system was powered before losing its $\pm 5\%$ accuracy.

-Current measurements processed by the microcontroller must match a reference ammeter measurement within $\pm 5\%$.

Incorporating calibration in the code and also thanks to the consistency of the current sensor we had purchased, the current values processed and calculated by the microcontroller matched current measurements of the ammeter within $\pm 5\%$. This only needed to be calibrated in the code once.

4.5. Display

-Pressing the button for less than 2 seconds will switch modes.

-The red LED is on and the green LED is off when the display is in Mode 1. The green LED is on

and the red LED is off when the display is in Mode 2.

-In Mode 2, holding down the button for 2 seconds resets the display to 0.

In the preliminary testing stages, we tested the display code with the Arduino Uno. We attached the 4-digit 7-segment display, LEDs, and button to a breadboard and assigned the 4-digit 7-segment display pins to the Arduino Uno for testing. We generated two sinusoidal waveforms and input them into the Arduino Uno, one to simulate the current and the other to represent the voltage after the signals had been processed by the sensing circuits. This allowed us to test the display code with mock values of current and voltage to see if the power was being calculated correctly, and also allowed us to test the functions of the display code with the button.

When the display code was tested on the bike, the display updated its values roughly every second so it was slow enough for the person to read the values. We verified that pressing the button for less than 2 seconds switched between Mode 1 and Mode 2, and when we held down the button for longer than 2 seconds when in Mode 2, the total energy was reset to 0. Holding the button down for longer than 2 seconds in Mode 1 did not reset the total energy. With these tests, the listed requirements above were all fulfilled in the final product.

5. Conclusion

5.1. Accomplishments

The first high-level requirement was that our system is able to generate energy. Due to feedback from the design review, we were advised to change our original goal of 40 W, so we did all of our testing at 10 W. Our project was able to successfully complete this requirement.

The second high-level requirement was to have a display system for the consumers that showed the power being generated and the energy generated in a time period.

The third high-level requirement was to have the power output from the bike synchronize to the gym's grid. Due to limitations with the microcontroller, we were unable to actually implement the algorithm with our PCB. However, we were able to show the algorithm working with the Arduino Uno. We showed our generated signal in about a 6 degree phase difference with the reference signal which falls under the 20 degree limit from IEEE 1547 [3].

We were also able to implement extra safety features such as a fuse and relay to shut down and maintain our system.

We were able to achieve all three of our high-level requirements, albeit the grid-synchronization algorithm not integrated into our system. We still deem this project to be a success, since we would be able to integrate it into our system if we use another microcontroller.

5.2. Uncertainties

Our microcontroller was too slow to run the sensing code, the display code, and the inverter without the PLL algorithm incorporated. As it can be seen in the figures below, the inverter output was distorted due to the code having to complete the sensing and display code before being able to run the inverter output.

We considered finding a function that could incorporate a multi-core functionality so that the inverter code and the sensing & display code could run in parallel on separate cores, but our microcontroller does not have multiple cores. We also tried multi-threading, which was switching between lines of two functions, but the output was still distorted. Because of the limited speed that the microcontroller could execute these programs, our PLL algorithm was not integrated into our final product.

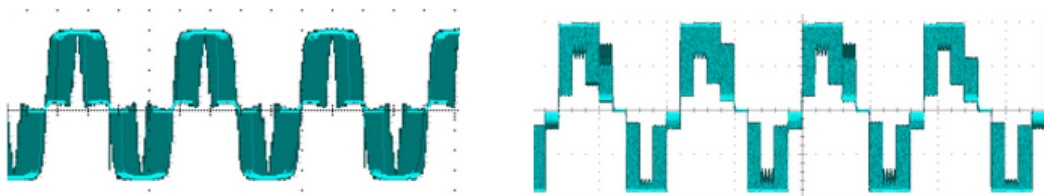


Figure 16 (left): inverter output without sensing & display code

Figure 17 (right): inverter output with sensing & display code

Also, our op-amps were very sensitive to temperature changes, thus not being able to scale and shift the inverter voltage appropriately for the microcontroller to process the actual

inverter voltage, and so not being able to accurately measure the power being generated. To resolve this issue, we can purchase op-amps that are more resistant to temperature.

In addition, our 15 VDC boost converter had a lot of ringing, which was mentioned in the Design Verification section. Because of this, we decided to use an external voltage supply for the safety of our system.

5.3. Future Work / Alternatives

We ran into delays in our project due to our microcontroller, so our most important advice for future work would be to carefully choose a microcontroller that is programmable with the Arduino interface, have more pins, and have multiple cores. The multi-core function would have allowed us to implement the synchronization code with the display code on our PCB. Since both of these programs do not rely on each other, they should be able to run on separate cores simultaneously. In addition, if there were more available pins on the microcontroller, we could have used an LCD display. This would have allowed for us to be able to display both the power and energy numbers on one screen rather than having to switch through modes, and also allow for more entertaining visuals for the consumer.

For our project, we did not connect our system to an outlet. Instead, we simulated a testing environment. For future work, if a group actually gets the approval to test out the system on the actual outlet of a gym, this would require consulting with grid synchronization professionals to implement additional requirements/features that would be needed. It is required for anyone sending power to the grid to consult their local utility companies.

Another thing to keep in mind for future improvements would be to account for higher voltages since we were limited to a lower voltage level. In order to account for the higher voltage, higher current, and higher power outputs, it is important to get higher current ratings for the components.

Another improvement would be to mitigate the heat accumulation. The PCB system got very heated during testing when we started pedaling very fast and it started smoking. To prevent this in the future, we would advise adding better heat ventilation for the box and heat sinks.

5.4. Ethical Considerations

In section 1.1 of the IEEE code of conduct, it states that we must “hold paramount the safety, health, and welfare of the public” [14]. The biggest concern for our project in terms of ethics and safety is the high voltages and currents. This is why we included multiple safety features into our system. We had the fuse which would blow if the current got too high, as well as the relay which would disconnect our system from the grid if no one is pedaling the bike. These precautions were put in place for our own safety and our consumers’ safety.

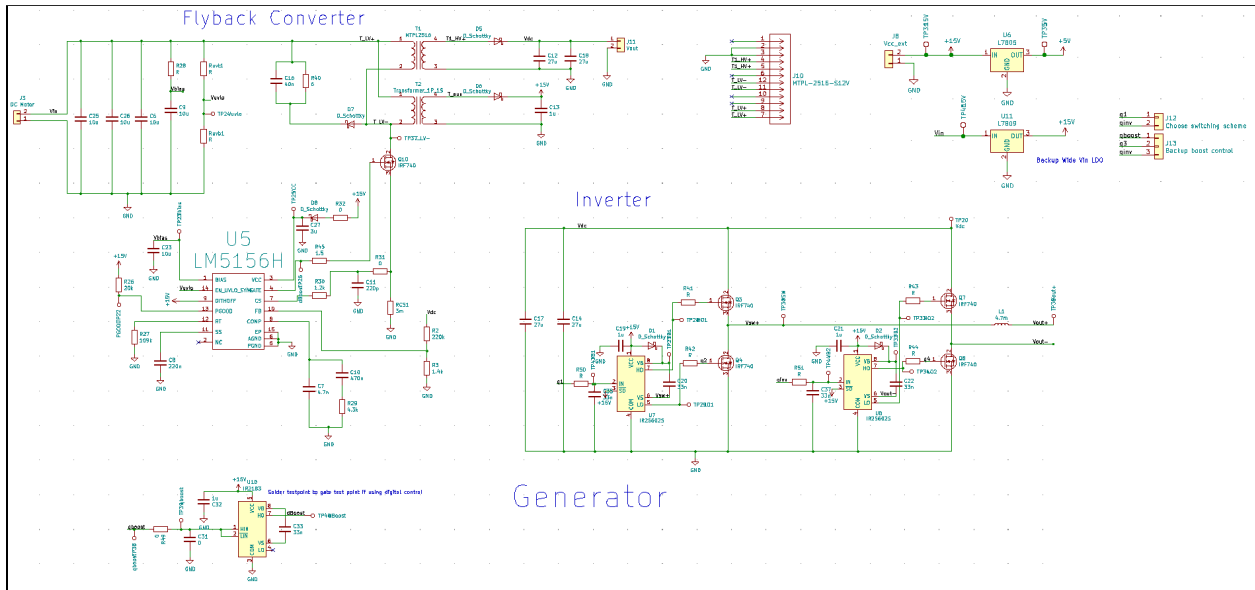
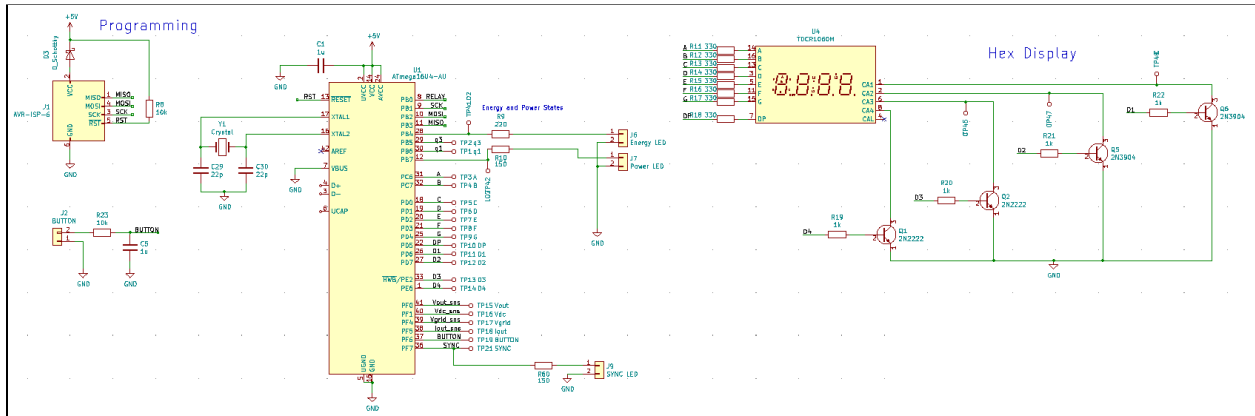
References

- [1] K. Popowych, "Power in Motion: Using Watts to Amp Up Indoor Cycling Drills" *AFAA*. [Online]. Available: <https://blog.afa.com/using-watts-to-amp-up-indoor-cycling-drills> [Accessed: 25-Sep-2021]
- [2] J. Patel, "TV Power Consumption in A Day?" *Lets Save Electricity*, 03-Jun-2020. [Online]. Available: <https://letsaveelectricity.com/how-much-power-does-a-tv-use-in-an-hour/>. [Accessed: 16-Sep-2021].
- [3] "IEEE Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces," in *IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003)*, vol., no., pp.1-138, 6 April 2018, doi: 10.1109/IEEESTD.2018.8332112.
- [4] "ATmega32U4 Datasheet," Atmel. [Online]. Available: <https://cdn.sparkfun.com/datasheets/Dev/Arduino/Boards/ATMega32U4.pdf>. [Accessed: 07-Dec-2021].
- [5] P. T. Krein, *Elements of power electronics*. New York: Oxford University Press, 2016.
- [6] R. Ridley, "Flyback Converter Snubber Design." *Switching Power Magazine*, 2005.
- [7] "LM5155/56 Flyback Controller Quick Start Calculator." Texas Instruments, 2020.
- [8] "Power Stage Designer." Texas Instruments Incorporated, Dallas, 2018.
- [9] "Non-Inverting Op-Amp Resistor Calculator" *Daycounter-Engineering Resources*. [Online]. Available: <https://daycounter.com/Calculators/Op-Amp/NonInverting-Op-Amp-Resistor-Calculator.phtml> [Accessed: 25-Sep-2021].
- [10] R. Teja, "Differential Amplifier" *Electronics Hub*. 21 April 2021. [Online]. Available: <https://www.electronicshub.org/differential-amplifier/> [Accessed: 25-Sep-2021].
- [11] G. Alonso, "LTSPICE: Voltage Controlled Switches," *Analog Devices*. [Online]. Available: <https://www.analog.com/en/technical-articles/ltspaceiv-voltage-controlled-switches.html>. [Accessed: 07-Dec-2021].

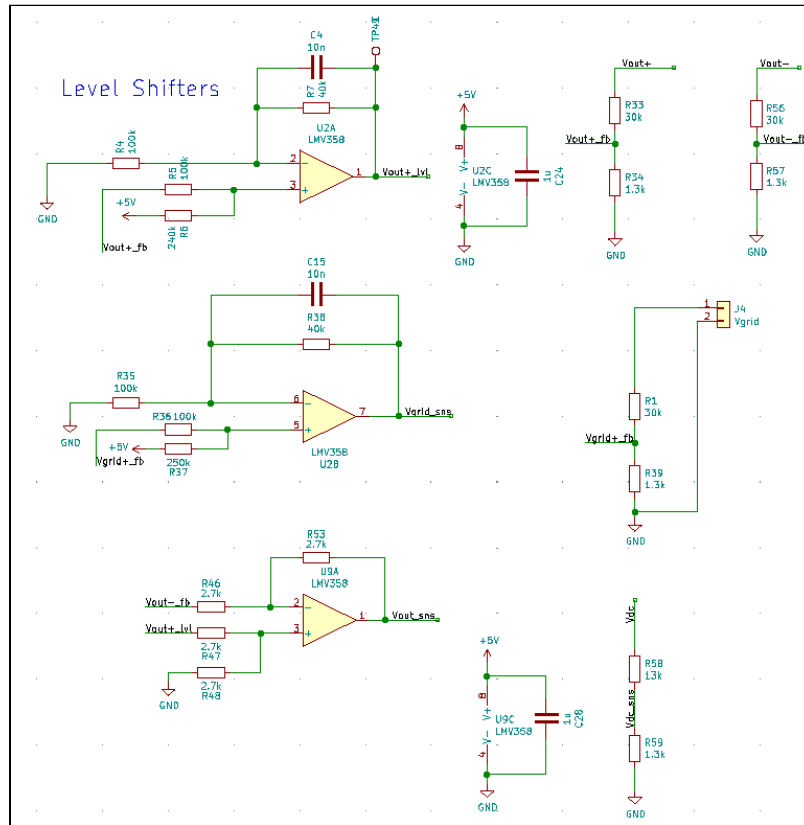
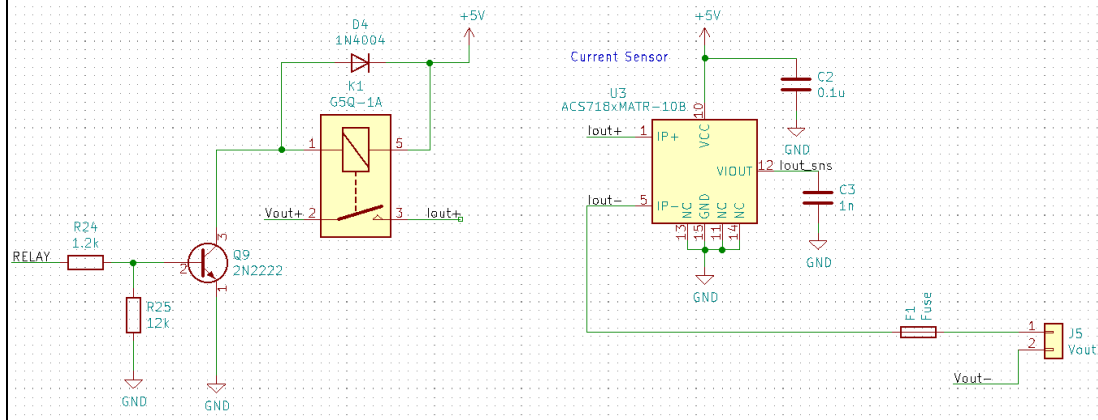
- [12] “Machine Shop” *School of Chemical Sciences at Illinois*. [online] Available:
<https://scs.illinois.edu/resources/cores-scs-service-facilities/machine-shop> [Accessed
25-September-2021].
- [13] “Salary Averages,” *The Grainger College of Engineering*. [Online]. Available:
<https://ece.illinois.edu/admissions/why-ece/salary-averages>. [Accessed: 25-Sep-2021]
- [14] “IEEE code of ethics,” *IEEE*, Jun-2020. [Online]. Available:
<https://www.ieee.org/about/corporate/governance/p7-8.html>. [Accessed: 16-Sep-2021]

Appendix A: Overall Schematic and Display Code Flowchart

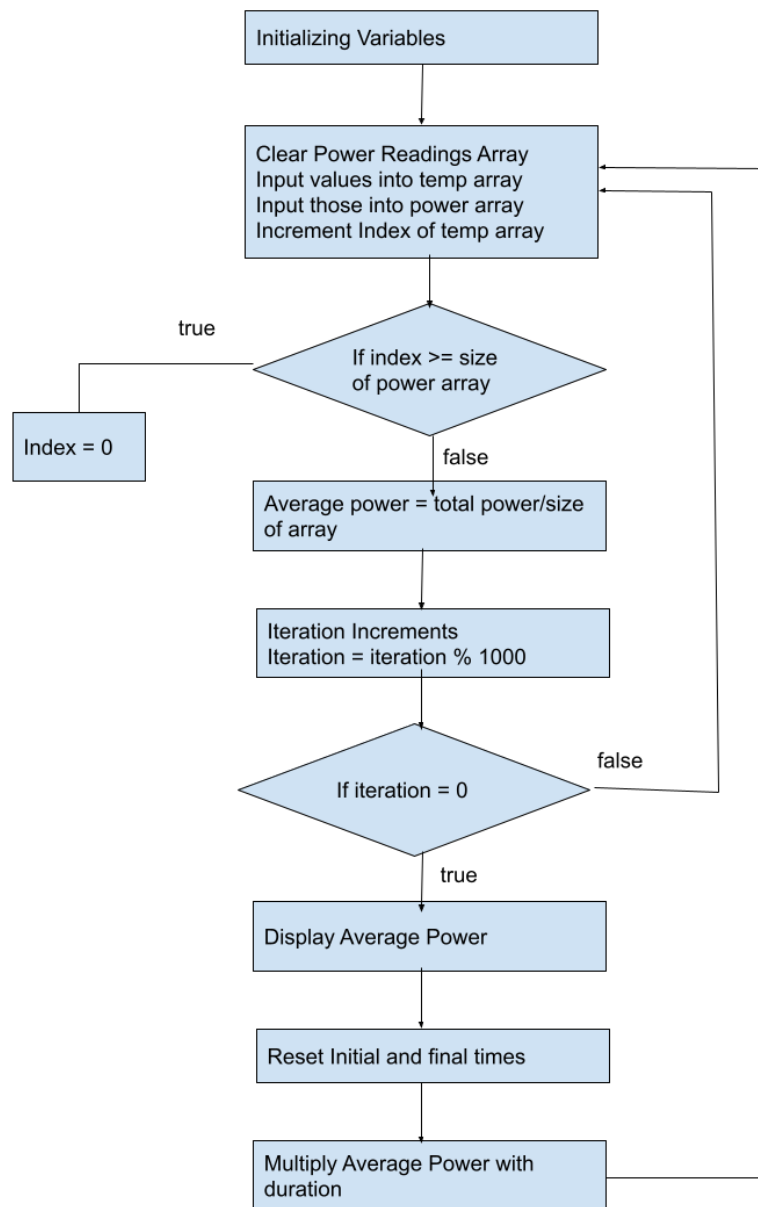
Schematics:



Grid Safety



Display Code Flowchart:



Appendix B: Parts & Costs Table and Schedule

Parts & Costs Table:

Part	Manufacturer	Part Number	Quantity	Unit Cost
Red 4-digit 7-segment display	Vishay Semiconductor Opto Division	TDCR1060M	x1	\$2.60
Microcontroller	Microchip Technology	ATMEGA32U4RC-AU-ND	x1	\$5.14
Op-amp (2 op-amps per chip)	Texas Instruments	LMV358QDDUR	x2	\$0.93
Current sensor	Allegro MicroSystems	ACS718KMATR-10B-T	x1	\$4.87
Transistor	NTE Electronics, Inc	2N2222 or BC337	x5	\$0.63
Fuse	OptiFuse	2298-TCC-3A-ND	x1	\$2.74
Fuse holder	Eaton - Electronics Division	PCB FUSE HOLDERS FOR 5 X 20MM FU	x1	\$4.32
Relay	Omron Electronics Inc-EMC Div	G5Q-1A4 DC5	x1	\$1.48
Wall plug	Leviton	3W101-00E	x1	\$2.98
Motor	Yaskawa Electronics	P12H-DB11	x1	~\$45
Spin bike (being retrofitted)	N/A	N/A	x1	~\$200
Boost controller	Texas Instruments	LM5156HPWPR	x1	\$2.28
Flyback transformer	Vishay Dale	MTPL-2516-S12V	x1	\$17.00
5V voltage regulator	onsemi	LM7805ACT-ND	x1	\$0.46
Switching MOSFETs (400V)	STMicroelectronics	STP7NK40Z	x5	\$1.63
Half-bridge driver	Infineon Technologies	IR25602STRPBF	x3	\$0.97
Second flyback transformer	Würth Elektronik	750315126	x1	\$3.28
Inverter output inductor	EPCOS - TDK Electronics	B82724J8482N040	x1	\$5.95
DC link capacitors	United Chemi-Con	EKXJ401ELL270MJ30S	x4	\$1.34
Electrolytic capacitor (10uF)	Nichicon	UPS2A100MED1TA	x1	\$0.43
Filter capacitor (1uF)	Würth Elektronik	860020672005	x1	\$0.10
Schottky diodes (600V)	SMC Diode Solutions	SK2C0A	x2	\$0.23

Note: Small and very cheap components that are easily obtainable– such as LEDs, buttons, surface mount resistors– are not included in the parts table.

Schedule:

Week	Elisa	Raihana	Tiffany
10/4	Finalize PCB design	Prepare for Design Review/order parts	Prepare for Design Review/order parts
10/11	Work on open loop inverter control	Work on display code for microcontroller	Work on sensor code for microcontroller
10/18	Debug microcontroller	Unit test safety and display, start soldering PCB	Debug microcontroller
10/25	Test inverter	Finish soldering PCB	Solder sensor components, unit test voltage sensors
11/1	Debug boost converter	Debug HEX display	Modify SevSeg library for HEX display
11/8	Debug boost converter	Test Grid Safety and Metering	Combine & debug components of Control and Metering
11/15	Integrate subsystems	Test complete system	Test complete system
11/22	Work on PLL algorithm	Work on presentation	Work on presentation
11/29	Put PCB into box and set up connections	Put PCB into box and set up connections	Put PCB into box and set up connections
12/6	Work on final paper	Work on final paper	Work on final paper

Appendix C: Requirements & Verification Tables

Striked-through text shows revisions to the original requirements & verification in the Design Document.

DC-DC converters

Requirements	Verifications	Requirement met?
DC-DC boost converter must be able to output +70VDC $\pm 5\%$. 19 VDC $\pm 5\%$	(a) Connect a DC voltage supply to the input of the DC-DC converter. (b) Connect a 100 MOhm resistor across the output of the DC-DC converter. (c) Connect an oscilloscope to the output of the DC-DC converter. (d) Slowly sweep the voltage generator from 5VDC to 35VDC. (e) Verify that the DC-DC converter outputs +70V $\pm 5\%$. 19 VDC $\pm 5\%$	Yes
DC-DC converter must generate a 15 V ± 5 V supply for the voltage regulators.	(a) Connect a DC voltage supply to the input of the DC-DC converter. (b) Connect a 100 MOhm resistor across the output of the DC-DC converter. (c) Connect the output of the DC-DC converter to an oscilloscope. (d) Slowly sweep the voltage generator from 5VDC to 35VDC. (e) Verify that the DC-DC converter outputs 15V ± 5 V for the entire input voltage range.	No

Inverter

Requirements	Verifications	Requirement met?
Inverter must be able to output $\pm 70/\sqrt{2}$ $\pm 19/\sqrt{2}$ Vrms $\pm 10\%$ with a power factor of 0.7 or greater.	<p>(a) Connect a voltage generator to the bike input.</p> <p>(b) Connect a load to the output of the inverter.</p> <p>(b) Connect a differential voltage probe across the inverter output load and to the digital power meter.</p> <p>(c) Place a current sensor across the inverter output load. Connect the sensor to the digital power meter.</p> <p>(d) Create a ± 70 ± 19 VDC signal from the voltage generator.</p> <p>(e) Verify that inverter output is $\pm 70/\sqrt{2}$ $\pm 19/\sqrt{2}$ Vrms $\pm 10\%$, and confirm that the power factor is greater than 0.7 using the digital power meter readings.</p>	Yes
Output voltage frequency must be able to synchronize to an external sinusoidal signal's frequency within $\pm 10\%$ and have no more than a 20° phase angle between the two.	<p>(a) Connect a voltage generator to the grid voltage sensor. Connect another voltage generator to the input of the inverter. Connect both voltage generators to the oscilloscope.</p> <p>(b) Connect an oscilloscope to the output of the inverter.</p> <p>(c) With the voltage generator connected to the grid voltage sensor, create a 50 VAC 60 Hz signal. With the voltage generator connected to the input of the inverter, create a 50VDC signal.</p> <p>(d) After 10 seconds or more (see Microcontroller Requirements and Verifications table), freeze the oscilloscope measurement. Measure the phase difference between the grid signal and the output of the</p>	<p>Partially.</p> <p>Not integrated with our board (code works separately with Arduino Uno)</p>

	<p>inverter using the oscilloscope.</p> <p>(d) Verify that the inverter output's frequency is within $\pm 10\%$ and has no more than a 20° phase angle compared to the AC voltage generator's signal.</p>	
--	--	--

Relay

Requirements	Verifications	Requirement met?
The relay disconnects the grid from our device within 2 seconds of inactivity of the bike.	<p>(a) Using a multimeter in Continuity mode, connect the probes across the relay.</p> <p>(b) Start biking. Check that the relay is connected while biking.</p> <p>(c) Open a count-up timer app on the phone.</p> <p>(d) Stop biking. Start the count-up timer. With the multimeter, verify that the relay becomes open within 2 seconds.</p>	Yes

Sensors

Requirements	Verifications	Requirement met?
Grid voltage measurements processed by the microcontroller must have an accuracy of $\pm 5\%$.	<p>(a) Connect a voltage generator to the input of the grid voltage sensor (voltage divider).</p> <p>(b) Create a 50 VDC 15 VDC signal using a voltage generator.</p> <p>(c) Attach the positive oscilloscope probe to the output of the op-amp circuit, and the negative probe to GND.</p> <p>(d) Upload code to the microcontroller that shows the grid voltage being interpreted and used by the microcontroller to be displayed on the display.</p> <p>(e) Verify that the display reading is $\pm 5\%$ of</p>	Yes

	the voltage generator.	
Inverter voltage measurements processed by the microcontroller must have an accuracy of $\pm 5\%$.	<p>(a) Add a 50-Ohm load to the inverter output. Attach a voltage sensor across the load, and connect the sensor to a voltmeter.</p> <p>(b) Connect a voltage generator to the bike input.</p> <p>(c) Upload code to the microcontroller that deactivates the inverter and makes the inverter's output into DC instead of AC. In addition, the code makes it so that the display shows the voltage being interpreted and used by the microcontroller.</p> <p>(d) Generate a 15 VDC signal with the voltage generator.</p> <p>(f) Verify that the display reading is $\pm 5\%$ of the voltage generator.</p>	No
Current measurements processed by the microcontroller must match a reference ammeter measurement within $\pm 5\%$.	<p>(a) Add a 50-Ohm load to the inverter output. Attach a current sensor across the load, and connect the sensor to an ammeter.</p> <p>(b) Connect a voltage generator to the bike input.</p> <p>(c) Upload code to the microcontroller that deactivates the inverter and makes the inverter's output into DC instead of AC. In addition, the code makes it so that the display shows the current being interpreted and used by the microcontroller.</p> <p>(d) Generate a 15 VDC signal with the voltage generator.</p> <p>(f) Verify that the display measurement is within $\pm 5\%$ of the ammeter's measurement.</p>	Yes

Display

Requirements	Verifications	Requirement met?
Pressing the button for less than 2 seconds will switch modes.	(a) Have the display be at Mode 2. The green LED should be on.	Yes
The red LED is on and the green LED is off when the display is in Mode 1. The green LED is on and the red LED is off when the display is in Mode 2.	(b) Start biking until the display shows that the energy generated is larger than 0. Remember this value for step (d).	Yes
In Mode 2, holding down the button for 2 seconds resets the display to 0.	(c) Stop biking. Press the button (for less than 2 seconds). Verify that the display changes to 0, the red LED turns on, and the green LED turns off.	
	(d) Press the button again (for less than 2 seconds). Verify that the display changes from 0 to the value at the end of step (b), the green LED turns on, and the red LED turns off.	Yes
	(e) Pull out a phone app count-up timer.	
	(e) Start holding the button down, and simultaneously start the phone app count-up timer. Verify that the display resets to 0 when the phone app reaches 2 seconds.	