ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Educational Wind Powered Charger

Team #23

ZIXI LI
(zixili2@illinois.edu)
MARIA CARRALLO ESCUDERO
(mariac7@illinois.edu)
LINGXIAO MOU
(lmou2@illinois.edu)

TA: Uma Lath

May 3, 2022

Abstract

This report detailed the design, implementation and testing results of a bike-mounted wind turbine generator. The system is designed to be installed on a bike. A motor is mounted to the front of the bike, and it connects to a PCB box mounted under the cyclists' seat, from which a USB port is provided for charging. The system also has an anemometer mounted at the front wheel for measuring wind speed, so we can achieve higher wind power harvesting efficiency. When the cyclist is riding, the system is able to charge its internal battery with wind power, and the battery provides power to the USB port so the user can charge their appliances.

Contents

1 Introduction

1.1 Project Description

The project focuses on designing a bike-mounted windmill. A wind turbine will be mounted at the front of a bike, and the design will harvest energy from the turbine when the cyclist is riding the bike. Having such a device can bring up people's awareness of renewable energy and energy conservation. Therefore, the project is named "Educational Bike-Mounted wind power generator".

1.2 Problem

- **Green Energy Education.** As more industrial companies turn to cleaner energy utilization, wind power has become one of the main power sources for energy supply. From an industry point of view, it would be beneficial to educate students about such technologies are used in real-world applications.
- **Provide an alternative power source for cyclists riding in remote areas.** Lots of cyclists prefer to ride in wild areas, such as mountains, grasslands, desert, and remote gravel tracks, where electricity is absent or scarce. A charging tool that can supply power to portable electronic devices, such as smartphones, GPS, camping lamps or satellite radios, would be beneficial. Such devices have a small battery but also a small power drain, making it possible to charge them with tiny power sources like a small wind turbine.

1.3 Solution

We are proposing a portable, inexpensive, easy-to-use bike-mounted wind turbine. This power harvest system should be efficient when riding a bike. Furthermore, it can provide charging capabilities to portable devices through a USB port.

1.4 High Level Requirements

- **Portable blade and rotor system.** We want to farm energy and protect our rotor as much as possible. When the wind is too high(20 mph) for the rotor to rotate, the rider has the choice to temporarily remove the rotor system and easily install it back.
- **The system is able to charge a phone.** The system should provide enough power output(5V 500mA) when the rider is riding the bike at 15 mph. When charging a phone, the phone should show charging status.
- **The entire system's width must not exceed 40 cm.** We do not want our wind turbine to be too large in diameter, which might disturb the view of riders.

1.5 Block Diagram

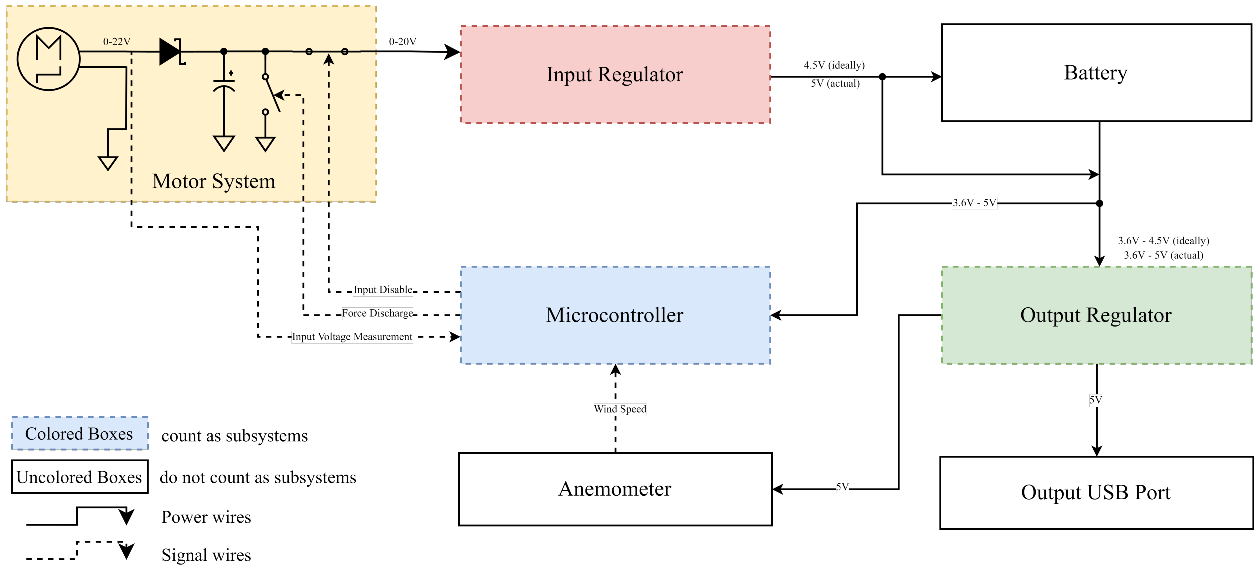


Figure 1: Design Block Diagram

Figure 1 shows the final block diagram of our design. The design stayed mostly the same as our original design, but the underlying implementation has changed significantly. To summarize, the original design featured an analog control system that adjusts the input more precisely and responds to input changes more actively, while the new method features a digital control that is easier to implement, less error-prone, and consumes less energy.

1.6 Block Descriptions

1.6.1 Motor System

This subsystem connects to the wind turbine and houses two switches. One switch disconnects the motor from the system and lets it spin freely; another switch connects the turbine to ground with a 100Ω resistor, drawing a relatively large amount of current from the motor and forcefully slowing it down. Switches are controlled by the microcontroller.

1.6.2 Input Regulator

This subsystem accepts unregulated voltage and converts it to a regulated $5V$ output for charging the battery.

1.6.3 Battery

This subsystem holds a protected battery module. It has protection against over-voltage and under-voltage.

1.6.4 Output Regulator

The output regulator converts battery power to $5V$ for the USB outlet.

1.6.5 Microcontroller

The microcontroller runs the MPPT algorithm and controls the motor system to achieve maximum power-point tracking.

1.7 Visual Aid

Figure 2: Visualization for the System.

Figure 2 shows the finished product. A motor with a small turbine is mounted in front of the bike, an anemometer is mounted to the side of the front wheel, and the PCB sits in the black box under the cyclists' seat.

2 Design

2.1 Design Procedure

The general idea of our design is to harvest the maximum amount of power from the motor system. Originally we thought about not using any battery and directly connecting the motor to a regulator, then to the output. However, after some considerations, this design would barely work.

Firstly, the power from the motor system is very small, but USB-powered devices, such as phones, require an adequate amount of current to recognize charging status. USB 2.0 specification required a minimum of $100mA$ for low-power hub ports and a minimum of $500mA$ for high-power hub ports[1, pp. 178, Table 7-7]. A USB charger is usually defined as a high-power hub port, but the motor maxes out at $1W$ during our testing. Even if our system is fully efficient, it will not be able to provide $5V$ $500mA$ output to other devices. When the appliance draws too much current, the regulator output voltage will drop out of the USB specified voltage range. Therefore, a battery must be included. A battery module can be charged at any charging current below its limit. When no device is plugged in, the system can charge the battery. When a device is plugged in, the system can charge the device using energy in the battery.

Secondly, the Voltage-Current characteristic of a motor is very special. It can roughly be modeled as a high-impedance voltage source, but that is not entirely the case. Under a specific wind speed, a motor has an optimal operating voltage, at which it can output the maximum amount of power[2]. However, an appliance plugged into the USB port is oblivious to the power source before the USB port, and it cannot control the amount of current drawn from the motor to achieve the best efficiency. Therefore, we need a Maximum Power-Point Tracking algorithm and circuitry to make the motor operate at the best operating point. The MPPT algorithm requires a pool that accepts the energy being generated, but an appliance plugged into the USB port cannot behave like a pool. Not only because the power is not high enough but also the amount of deliverable power is going to be fluctuating due to the implementation of MPPT. If the appliance detects that the voltage drops too much when current is being drawn, the appliance may stop drawing current from the USB port. In such cases, no current is being drawn from the motor through the voltage regulator, so the rotation speed of the motor is not slowed down, the output voltage of the motor will therefore not drop to optimal voltage, causing the MPPT execution to fail and wasting energy. Therefore, we need a pool that can absorb any amount of power delivered to it, and that pool could be a battery.

Therefore, a battery is essential in the system, and the system focuses on charging the internal battery at maximum efficiency. A lithium battery typically outputs $3.7 - 4.2V$ of voltage, making an output regulator necessary. This regulator should convert battery voltage to $5V$ to meet the specifications of USB.

Charging a lithium battery is common practice. Usually, the power source used to charge a single-cell lithium-ion battery is set at $4.4V$ maximum, with an output current within the limits of the battery. When the battery is not fully charged, the battery will draw all the available current, and the voltage from the power source drops to a level slightly higher than the battery voltage. During the process, the voltage across the battery rises. When charging is complete, battery voltage rises, gets closer to the maximum supplied voltage, draws less current, and eventually stops when battery and supply voltages match, and the power source is no longer able to provide current to the battery. Originally in the design, we wanted to build a $4.5V$ regulator as the input regulator and add a Schottky diode to prevent reverse current, as well as deducting a voltage drop. However, the variable output version of the control IC we use, `TPS63070`, was out of stock. We considered other solutions, such as using two regulators for step-up and stop-down respectively and use a comparator to decide which regulator to enable. However, adding a comparator requires another power system to power the comparator, which complicates the design and increases the power consumption of the whole system. We decided to use its fixed-output variant, `TPS630701` to substitute for it, which has a fixed output of $5v$. When the battery is not fully charged, it can still draw current from the regulator and drag its output voltage down to the current battery level, making the regulator output behave like a current source. The battery protection board detects overvoltage and cuts off charging when the battery is fully charged.

Then it comes to the design of MPPT. Beforehand, the optimal operating voltage of the wind turbine at different wind speeds will be measured with the use of electronic loads, and the windspeed-voltage relationship will be fit into a curve and programmed into the microcontroller. When the system is running, the microcontroller measures the wind speed, checks the curve and calculates the best operating voltage. If the current motor voltage is higher than the optimal voltage, then the microcontroller instructs the circuit to draw current from the motor, slow it down, and drop the voltage back to optimal voltage. If the current motor voltage is lower than optimal voltage, then the microcontroller instructs the circuit to disconnect the motor, so no current is being drawn and the motor will speed up, ramping up its output voltage. Apart from MPPT, when the wind speed becomes too high and the circuit cannot sink enough current to slow the motor down, the microcontroller should also be able to sink current from the motor in other means. Therefore, the motor subsystem is designed to have two switches. One switch is for connecting or disconnecting the motor, and another for forcefully drawing current from the motor and slowing it down. The motor subsystem also has voltage dividers, dividing the input voltage for the microcontroller to measure.

The microcontroller should have plenty of I/O pins for taking inputs from the anemometer, measuring input voltage, and controlling the motor. It is also preferred for the microcontroller to draw little power since the power draw of the MCU is an important component in the standby operating power of our system. Therefore, we chose our microcontroller to be `ATTiny84V`. It can operate directly from battery voltage and consumes little power. It also has power-off modes that we can use to save power.

To conclude, the final design of our system is what we presented in Figure 1 - A battery with input and output regulators, the motor system, the microcontroller, and an anemometer.

2.2 Design Details

2.2.1 Motor System

Motor and Blade Selection For the blade, we chose a three-blade turbine with blade length $R = 22cm$. The selection comes from assuming normal air density $\rho = 1.2kg/m^3$ (under $T = 20°C$) and wind harvesting efficiency $\eta = 40\%$ (the optimal wind conversion efficiency is 0.59, but real world limitation should be considered). The power could be generated is calculated to be [3]

$$P = \frac{\pi}{2} \cdot r^2 \cdot v^3 \cdot \rho \cdot \eta = 11.003W \tag{1}$$

For choosing the generator, using the blade above, we have tip speed ratio $TSR = 1.5$, assuming an apparent wind speed of $v = 6.7m/s$ (15mph), we can calculate the rotation speed in rotation per minute (RPM) with the following equation.[4]

$$60 \cdot v \cdot TSR/(2\pi R) = 60 \cdot 6.7 \cdot 1.5/(\pi \times 0.44) = 440.14RPM \tag{2}$$

And under extreme conditions, such as a 60mph of wind ($v = 26.8m/s$), $r = 1746.38RPM$. The final result is less than the rated 3500 rpm of our motor, which means our turbine will not be broken under such conditions.

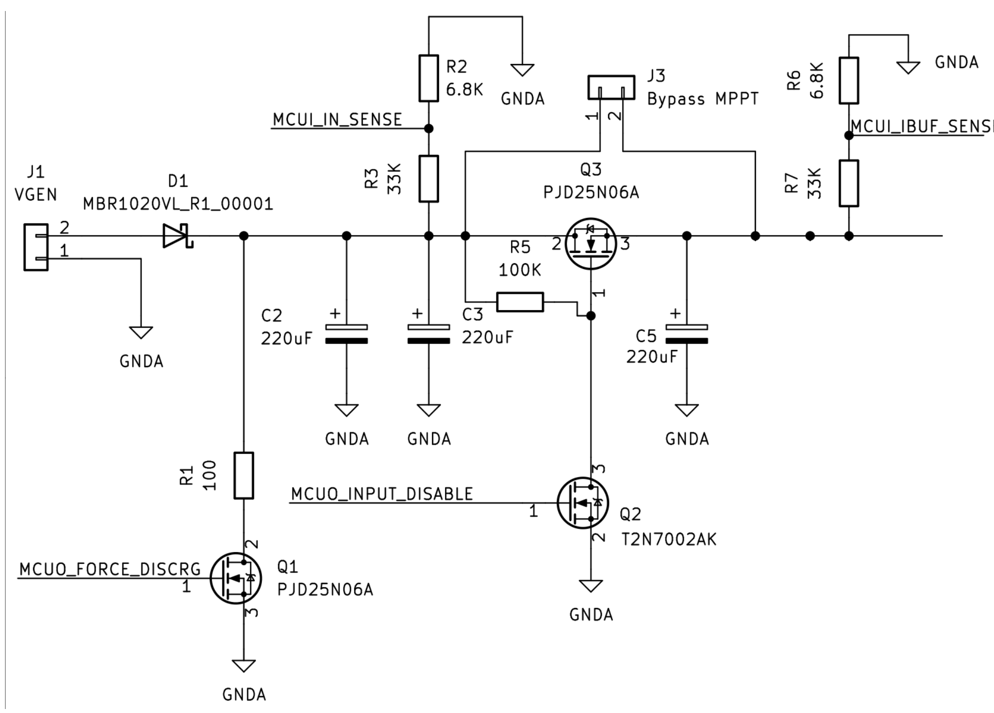


Figure 3: Motor Subsystem

System Specification Figure 3 shows the design of the motor system. Under best operating points, the power from the motor has a relatively higher voltage ($> 10V$) but a low current. MOSFET Q1 handles discharging of the motor, and the resistor R1 controls the discharge rate. Resistor R1 has a larger package and footprint that allows it to dissipate heat. MOSFET Q2 is a small-signal MOSFET that prevents the high pull-up voltage from being fed into the microcontroller, and MOSFET Q3 acts like a switch. Q1, Q2 and Q3 all have a V_{DS} breakdown voltage of $60V$, well above the maximum voltage the motor can generate under 60mph of wind. The voltages before and after the MOSFET switch are both measured by the microcontroller through voltage dividers.

When the signal MCUO_INPUT_DISABLE is set high, the motor is effectively disconnected from the system. When the signal MCUO_FORCE_DISCHARGE is set high, current is forcefully drawn from the motor and slows it down.

2.2.2 Input Regulator System

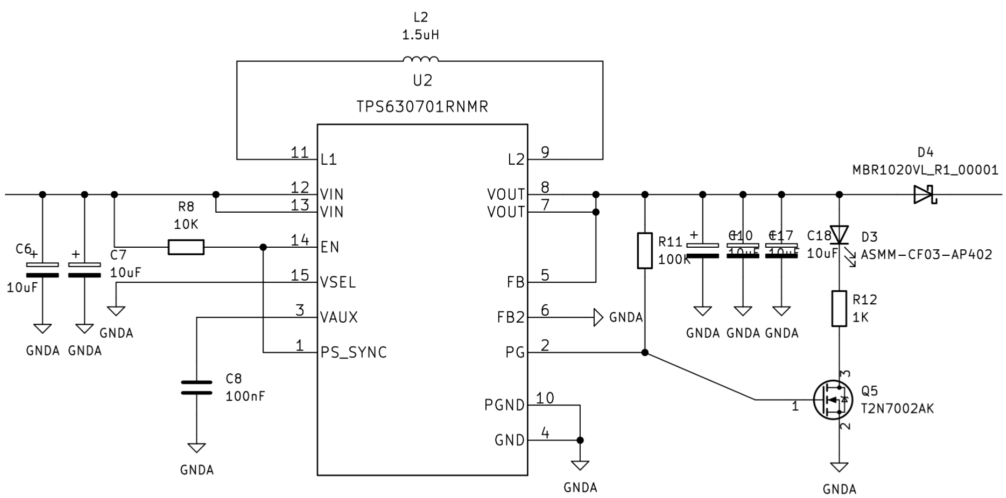


Figure 4: Input Regulator Subsystem

Figure 4 shows the schematic of the input regulator. The IC TPS630701RNMR is a buck-boost DC/DC converter IC with built-in MOSFETs[5]. We chose to use buck-boost architecture since it allows a wide input voltage range and provides good efficiency, and we chose this IC with built-in MOSFETs because it avoids having complicated, multi-layered PCB layouts, and it is designed to work in low-power applications.

The choice of inductor L2 is important for safe and efficient operation of the circuit. To make the circuit efficient, coil inductors are preferred over chip inductors because the chip inductors have a higher core loss. The inductor value determines the ripple current and load transient response. The larger the inductor is, the lower the ripple current will be, and the slower the transient response of the output becomes.

To calculate the peak current that goes through the inductor, we used the following equation:

$$DutyCycle = \frac{V_{out} - V_{in}}{V_{out}}$$

$$I_{PEAK} = \frac{I_{out}}{\eta \cdot (1 - DutyCycle)} + \frac{V_{in} \cdot DutyCycle}{2 \cdot f \cdot L}$$

The values are chosen as follows:

- $V_{OUT} = 5V$.
- $V_{IN} = 3V$. Lowest bound of input, when peak current is highest.
- $\eta = 0.7$. Estimated converter efficiency. Lower efficiency causes higher peak current. We used 70% to be safe.
- $I_{out} = 1A$. Estimated output current. Since the motor output never exceeded $1W$ when measuring with electronic load, the current should not exceed $200mA$. We used $1A$ to leave room for exceptions.
- $f = 2.1MHz$. Switching frequency of the controller. The operating frequency of the controller is $2.4MHz$, but it may slow down under low power operations. A lower frequency leads to a higher peak current. We chose $2.1MHz$, the lowest frequency it operates provided in the datasheet.
- $L = 1.5\mu H$. A recommended, balanced value as provided in the application information.

The resulting value is $I_{PEAK} = 2.57143A$. We chose inductor `SDR0805-1R5ML` that has a max DC current of $6A$ and saturation current of $9.1A$ to satisfy the requirements[6].

2.2.3 Battery

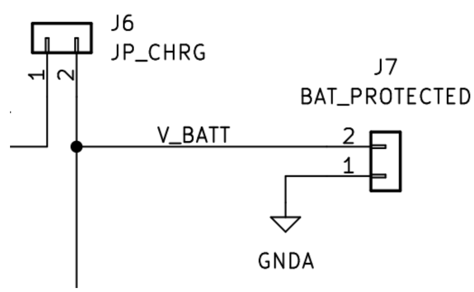


Figure 5: Battery (Connector)

As shown in Figure 5, there is a connector that connects to a battery on the PCB. `JP6` is a jumper for debugging, and `JP7` is the connector to the battery. The system requires a single-cell lithium-ion battery that can charge at $200mA$, discharge at $1A$, and can handle $1A$ of charging current in cases of exception. The product from Tenergy satisfied our needs.[7] It is rated for $520mA$ continuous charging, $1.3A$ rapid charging, and $3.5A$ discharging. The capacity is not a very important factor in our choosing, as long as it is not so small that gets saturated quickly (roughly $500mAh$ minimum). This battery has a capacity of $2600mAh$.

2.2.4 Output Regulator System

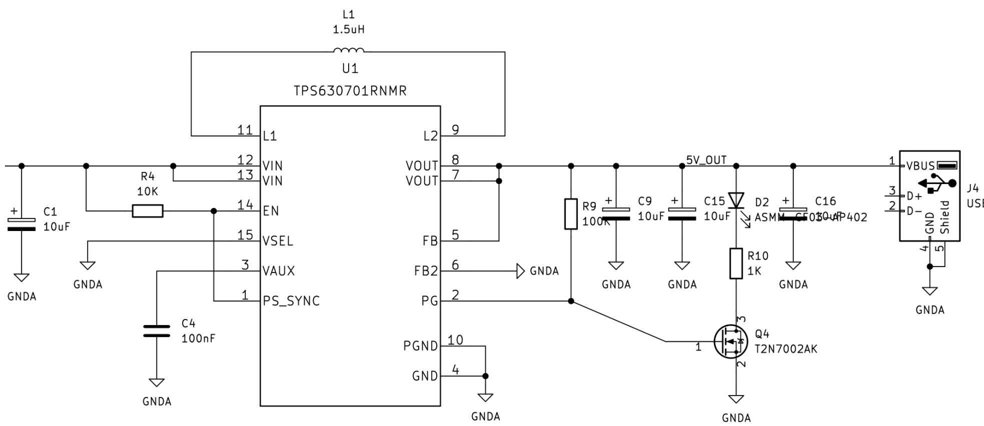


Figure 6: Output Regulator

As shown in Figure 6, the output regulator is the same design as the input regulator since the requirements are the same. There are two differences: There are fewer capacitors at the input since the input comes from a battery and is not expected to fluctuate significantly. There is no diode at the output, so there is no voltage drop.

2.2.5 Microcontroller System

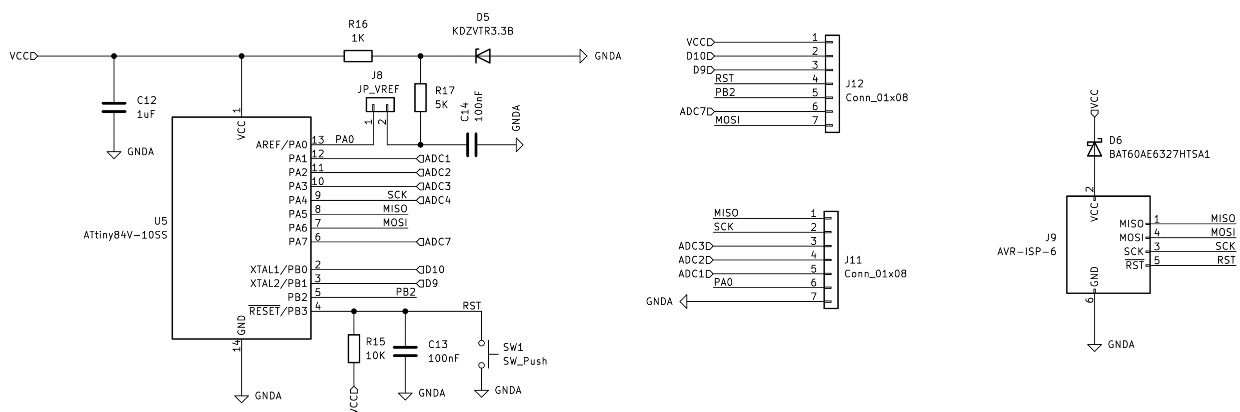


Figure 7: Microcontroller

Figure 8 shows the high-level connections between the microcontroller and anemometer, and Figure 7 shows the wiring of the microcontroller itself. The microcontroller can be powered from two power sources: It can be powered from the battery, in which case an external $3.3V$ voltage reference is provided for ADC operations. It can also be powered from the 5V bus, so no external voltage reference is needed. All analog inputs are designed to saturate at $3.3V$ for compatibility with both modes of operation.

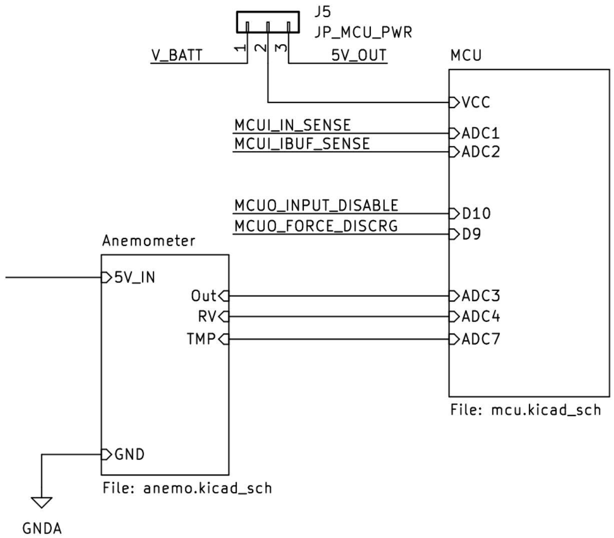


Figure 8: High-Level connections

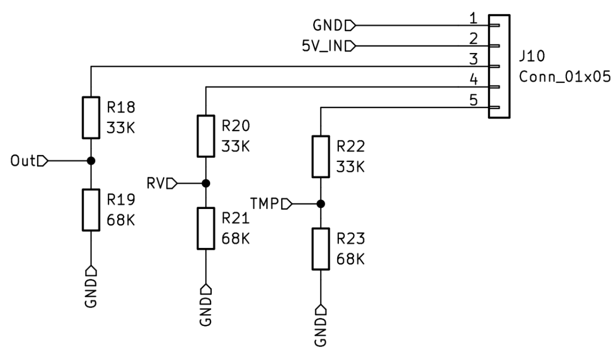


Figure 9: Anemometer connector

Anemometer The anemometer is a thermal anemometer based on the "hot-wire" method. The anemometer will heat a wire, and wind blowing onto the wire will cool it down. Therefore, the resistance of the wire changes with respect to wind condition. When the wind is fast, the temperature of the wire drops and its resistance drops, producing a voltage difference that can be amplified and measured.

It provides three signals: `RV` stands for raw value, which is not used. `TMP` is a temperature sensor that adjusts the reading with respect to ambient temperature, and `Out` is the reading. All three signals are sent to the microcontroller for calculation.

MPPT Algorithm At any specific wind speed, the Voltage-Power relation of the motor output reaches its peak at a certain voltage and becomes zero at high or low voltages. Y.Errami et al. investigated this relationship and acquired the following graph.[2]

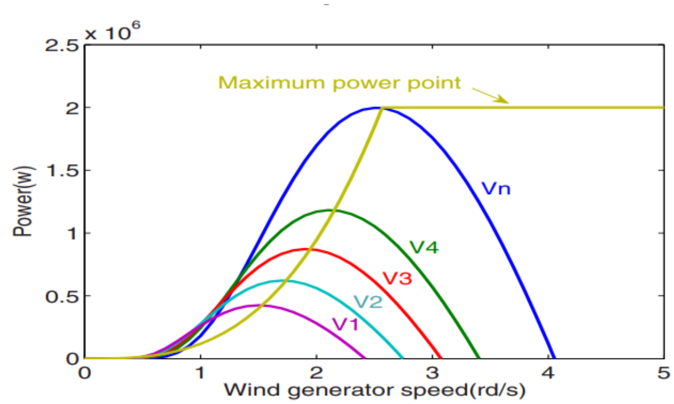


Figure 10: Maximum Power Point curve

In Figure 10, Wind generator speed is directly related to the output voltage of the motor and curves V1...Vn represents different wind speeds, where $V1$ is the lowest and Vn is the highest. It is obvious that seeking the maximum power point is essential in achieving higher efficiency.

Firstly, we need to measure the best operating voltages under different wind speeds for the specific turbine we are using. The optimal voltage at each wind speed is measured with an electronic load - The turbine is driven with a box fan, and the output terminals are connected to the electronic load. The electronic load is placed in constant voltage mode, and the voltage is slowly swept to find when the power becomes maximum. The process is repeated for multiple wind speeds and the result is fit into a curve.

The algorithm is as follows. When equilibrium is reached, the motor's coulomb force, wind force, and friction are balanced. Drawing current from the motor generates coulomb force inside the motor in the reverse direction where the motor is spinning, slowing it down. To maintain the motor around the best spinning speed, the only thing we need to do is to adjust the current being drawn, so the voltage stays optimal. The microcontroller measures wind speed, calculates the optimal operating voltage, and compares it with the current motor output voltage. If the current motor voltage is high, then we enable the input and draw current. As a result, the motor slows down, the output voltage is decreased and current is increased. If the motor voltage is too low, then we disable the input, so the motor speeds up and voltage increases. This method is not absolutely precise and responsive because it takes time for the microcontroller to react, but as seen in Figure 10, we will be able to let the motor operate in a range that has higher efficiency.

Figure 11 and 12 shows a simulation under the same conditions, with or without using MPPT. As it is shown in Figure 11, when the voltage of the motor is higher than best voltage, the device draws current from it in order to charge the battery. Figure 12 shows the same system without using MPPT and the load is always being connected to the motor. With the MPPT the system it was able to gain 437522.43 mJ and without it only 275612.73 mJ. Therefore, it is clear that implementing MPPT will bring a performance improvement to our project.

2.3 Requirements & Verification

The requirements and verification tables can be found at Appendix A.

2.3.1 Motor System

The motor subsystem is tested together with the input regulator since we already know the input regulator is working. There is an LED at the output of the input regulator, indicating whether power has been fed. Voltage and current readings came from the power supply. Table 1 shows the testing results of the motor subsystem. As shown, the `MCUO_FORCE_DISCRG` and `MCUO_INPUT_DISABLE` signals are working as expected.

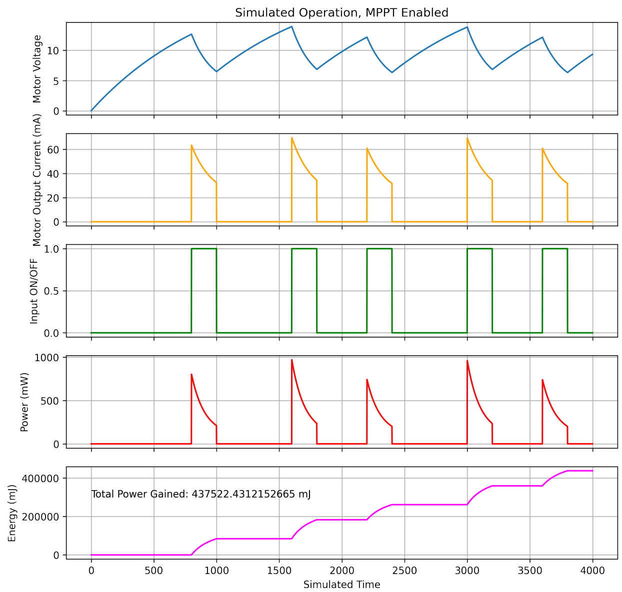


Figure 11: Simulation with MPPT

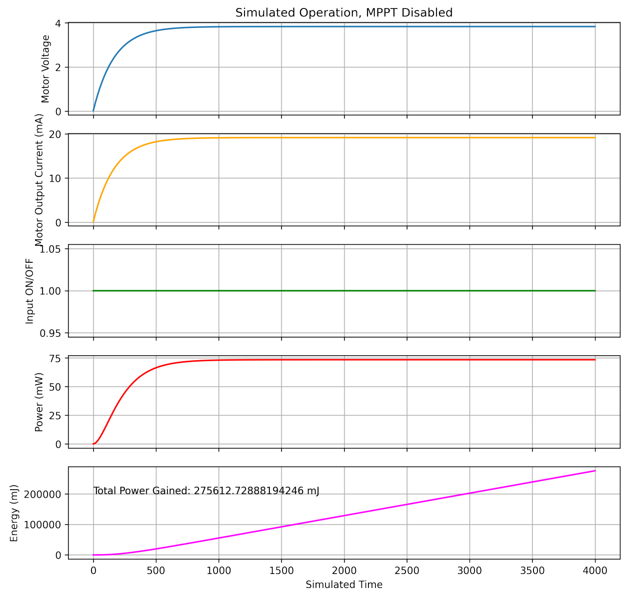


Figure 12: Simulation without MPPT

Input Voltage(V)	Input Current(A)	FORCE_DISCRG	INPUT_DISABLE	LED Status
12.00	0.001	LOW	LOW	ON
12.00	0.000	LOW	HIGH	OFF
5.14	0.050	HIGH	HIGH	OFF
12.00 (Reversed)	-0.000	LOW	LOW	OFF

Table 1: Motor Subsystem Testing Results

2.3.2 Input Regulator

For the input regulator, we care about its start-up time and efficiency. Due to the MPPT algorithm, the regulator will be turned on and off, so the start-up time of the regulator is important for MPPT to function correctly. The microcontroller runs its algorithm once per $1000ms$; we wanted the regulator to start within $100ms$ from when the input becomes available.

Figure 13 shows the start-up performance of the input regulator. It started within around 60ms, which meets our criterion.

Figure 14 shows the output voltage drop. The regulator meets the requirement to output 5V and is able to maintain $300mA$ of output current. The voltage drop is expected due to the diode, and it is within tolerance. The functionality will not be harmed as long as the output voltage is above $4.4V$.

Figure 15 shows the efficiency. The regulator's efficiency failed to meet the 80% target we

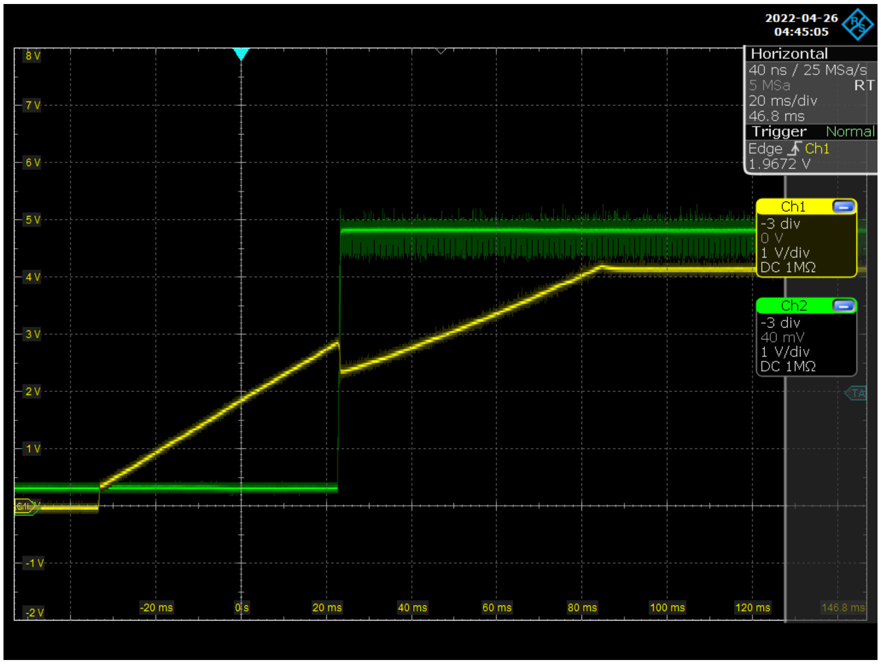


Figure 13: Input Regulator, Startup

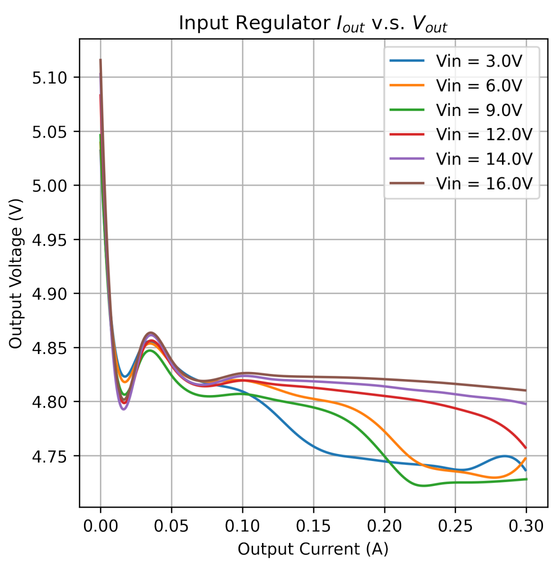


Figure 14: Input Regulator, Voltage Drop

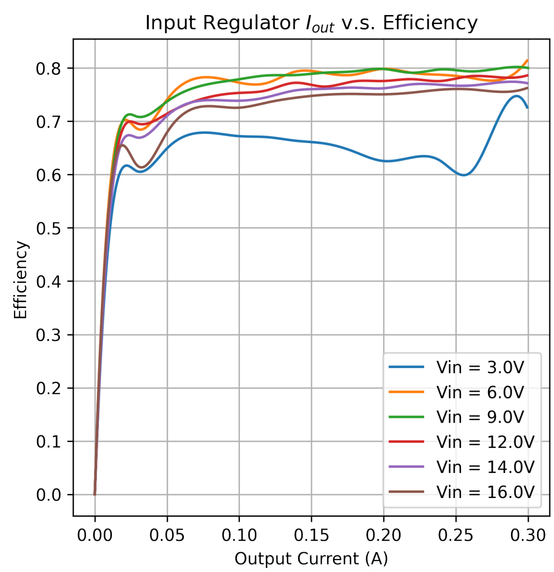


Figure 15: Input Regulator, Efficiency

set in our requirements, but it is very close to the target. The loss of efficiency partially comes from the inclusion of a diode and an LED indicator connected to the output that could not be turned off and not included in power measurement.

Figure 16 shows the output ripple when $300mA$ of current is being drawn. The amount of ripple stays within tolerance, and the voltage is always sufficient to charge the battery.

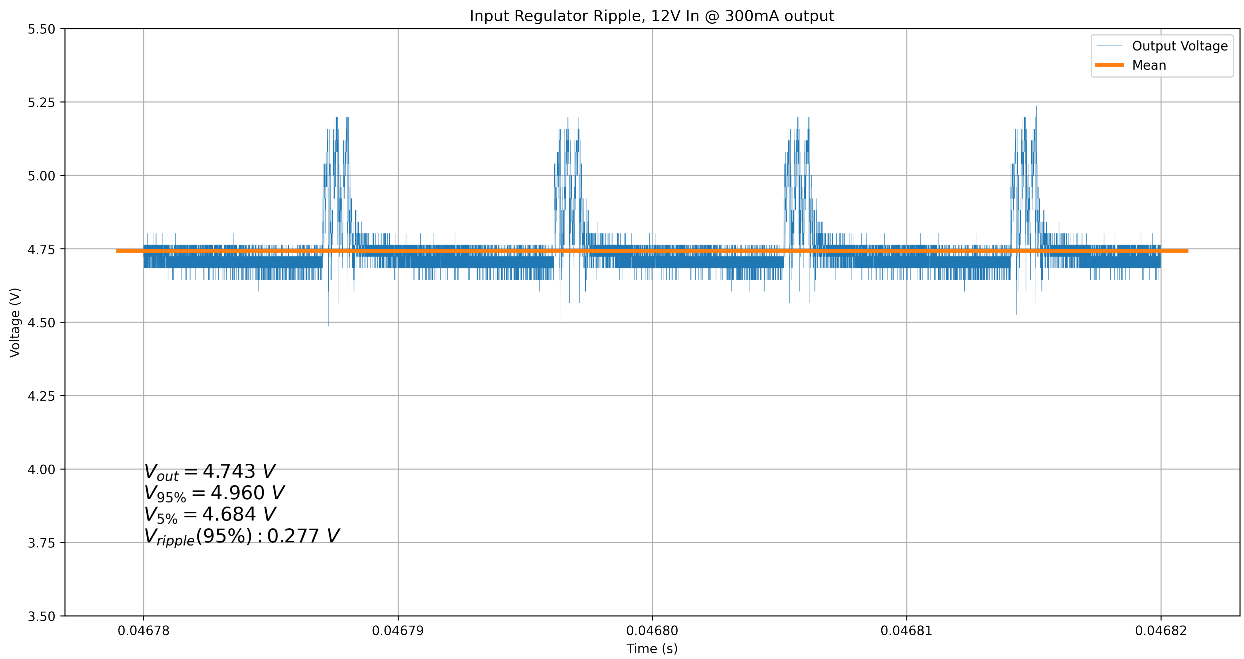


Figure 16: Input Regulator, Ripple

2.3.3 Output Regulator

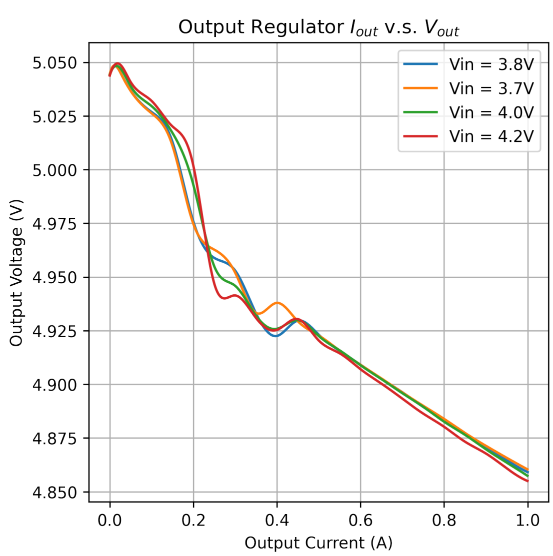


Figure 17: Output Regulator, Voltage Drop

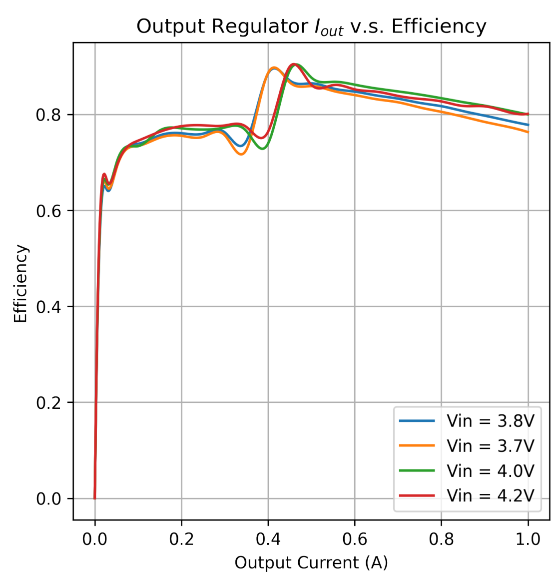


Figure 18: Output Regulator, Efficiency

Figure 17 and 18 shows the output voltage drop and efficiency when running at different input voltages. The voltage drop of the output regulator is lower than the input regulator due to the lack of diode, and the efficiency is also higher, especially at higher output currents. We used the USB port to charge a phone.

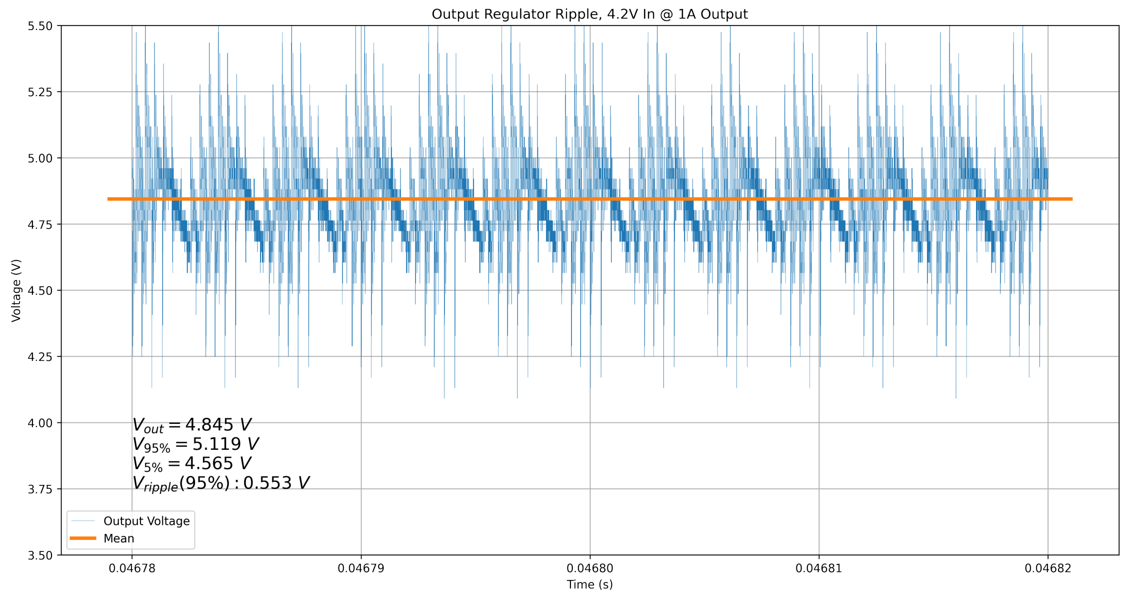


Figure 19: Output Regulator, Ripple

Figure 19 shows the output ripple. As we can see, the ripple voltage is higher than desired when delivering $1A$ of current. Efforts should be made to reduce output ripple, such as choosing a larger inductor for the H-bridge in buck-boost design, adding inductors in series of the output, or adding more capacitors to the output.

2.3.4 Microcontroller and Anemometer

The code used for the microcontroller can be found in Appendix D.

Before the system assembles, the best operating voltages under different wind speeds is measured as follows. The full testing data can be found at Appendix E.

1. Mount the wind turbine and anemometer to a table clamp. Connect wind turbine to electronic load. Connect the anemometer to the microcontroller.
2. Put the box fan in front of the turbine. Turn it on.
3. Record the anemometer output voltage and wind speed.
4. Set the electronic load to constant voltage mode. Sweep the voltage from 9.5V to 13V with a 0.5V step. Record current and power for each data point.
5. Move the location of the fan or change the gear setting of the fan. Repeat from Step 2.
6. After a sufficient number of wind speeds are tested, create a trendline with the best voltage for the generator depending on the output voltage of the anemometer, as seen in Figure 20

The microcontroller calculates the best operating voltage based on the anemometer input, based on the data measured with the steps shown above.

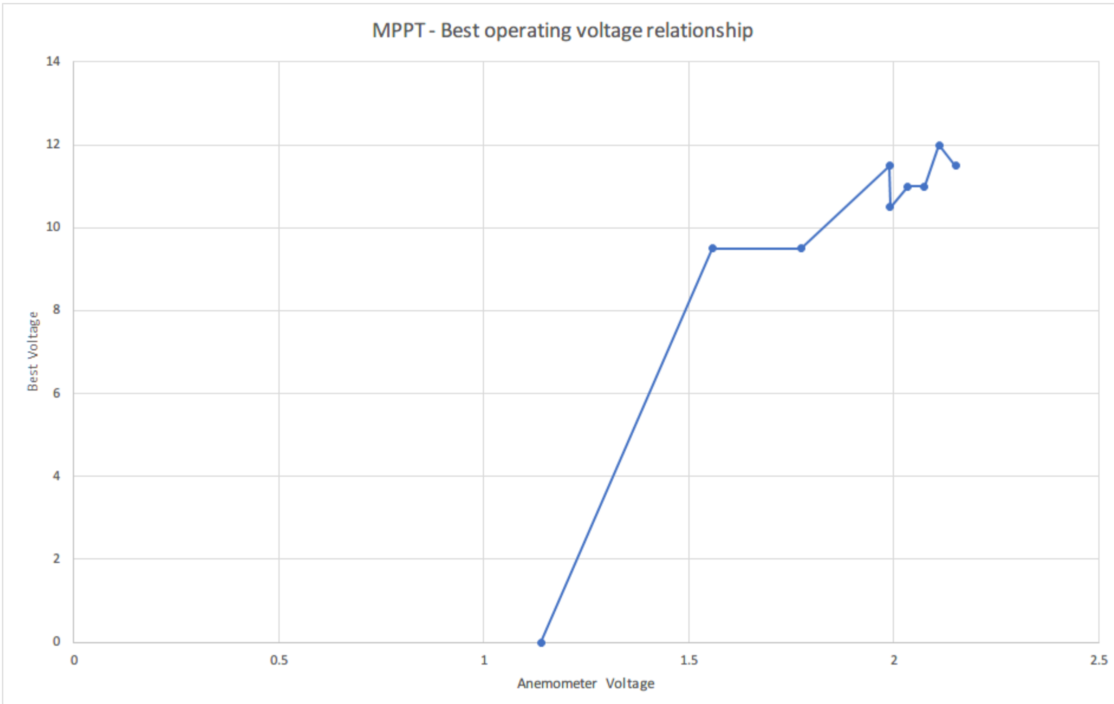


Figure 20: MPPT - Optimal Operating Points

As seen in Figure 21 and Figure 22, the trend line was divided into two parts due to a very different slope of the curve. The initial value for the graph is the output voltage provided by the anemometer when there was no wind, and as the wind speed increases, the voltage from the anemometer also increases.

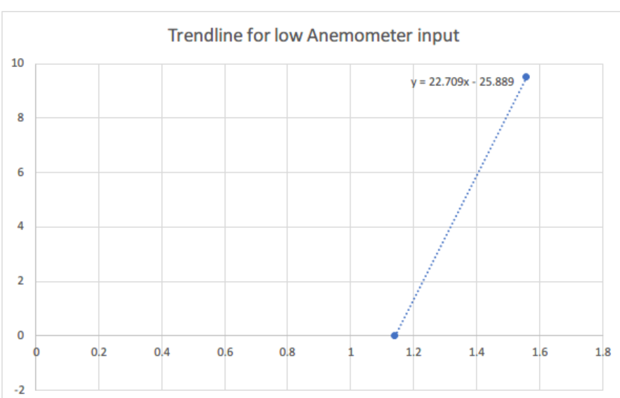


Figure 21: Trendline for low wind input

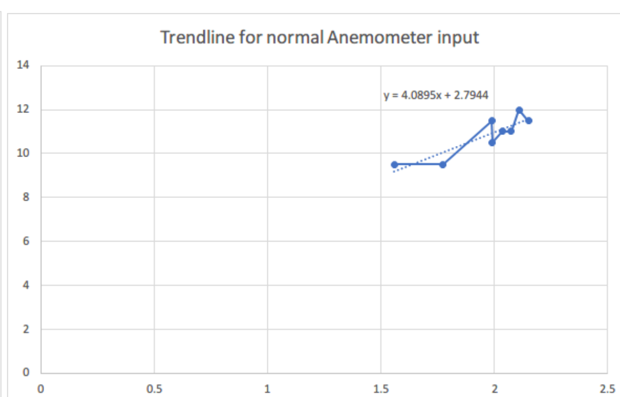


Figure 22: Trendline for normal wind input

The overall verification of the microcontroller system is done with the anemometer and battery plugged in, and the whole process was recorded. The anemometer was mounted on the table and blown with a fan. Table 2 shows two points we extracted from the video and serial console. We can conclude the system is working.

Anemometer Out	Best Voltage	Input Voltage	INPUT_DISABLE
2.454V	12.83V	12.02V	HIGH
2.261V	12.04V	13.02V	LOW

Table 2: MCU Control Outputs

2.4 Cost

2.4.1 Labor

The calculation is based on the average salary of a newly graduated electrical and computer engineering student, $ 91,781 per year. A year contains an average working time of 2080 hours, assuming an employee works 40 hours a week during 52 weeks. Therefore, the hourly income is approximately $ 44/hr.

Considering that each team member should spend 10 hours a week on the project in a 16 weeks semester, this would be a total of 160 hours to complete the project. Labor per member = 44 * 2.5 * 160 = $ 17,600. For the 3 members of the team, the total labor cost would be $ 52,800.

2.4.2 Parts

Shown in Table 3 is the cost of major components we purchased for this project.

Table 3: List of purchased parts

Item	Quantity	Vendor	Price/unit	Total Price	Date
RIYIN DC Motor Wind Turbine Generator	1	Amazon	$25.95	$25.95	April. 12
TPS630701RNMT	3	Mouser	$3.39	$10.17	April 11
BQ29707DSET-Battery Control IC	2	Mouser	$1.25	$2.5	April 11
ATTINY84-20PU-Microcontroller	1	Mouser	$2.72	$2.72	April 11
Resistors.	23	Digikey	$0.1 - $0.87	$15.42	April 11

Capacitors.	24	Mouser/Digikey	\$0.1 - \$1.58	\$18.85	April 11
Connectors.	12	Mouser/Digikey	\$0.26	\$3.12	April 11
Diode.	10	Mouser	0.39\$- 0.5\$	\$3.64	April 11
Inductors.	3	Mouser	\$0.8	\$2.4	April 11
MOSFET.	9	Mouser	\$0.61 - \$1.97	\$9.01	April 11
Mounting Hardware	1	Machine shop		\$200	-
Anemometer	2	Moderndevice	\$23.43	\$46.86	Mar. 1
Total	91	/	/	\$340.64	/

2.4.3 Schedule

Table 4: Schedule

Date	Zixi	Lingxiao	Maria
Feb 24 – March 2	PCB design	PCB design	PCB design
March 3 – 10	PCB redesign	Machine Shop Parts Ordering	Machine Shop Parts Ordering
March 11 – 17	Soldering and assembling	Test MPPT	Soldering and assembling
March 18 – 24	Testing (PCB redesign)	Assembling motor	Assembling blades
March 25 – 31	Test microcontroller	Test motors	Test microcontroller
April 1 – 7	Coding IC	Coding IC	Coding IC
April 8 – 13	Test PCB	Track MPPT	Soldering PCB
April 14 – 20	Mock Presentation and PCB testing	Mock Presentation and MPPT algorithm implementation	Mock Presentation and MPPT Data Gathering
April 21 – May 4	Finish Document	Finish Document	Finish Document

Table 4 shows our schedule for the project.

3 Conclusion

3.1 Accomplishments

Overall, all high level and low level requirements have been met. The wind power charger is capable of charging the battery under normal wind situation. In terms of the MPPT algorithm, the digital MPPT control is used to combat the low power output problem of our motor generator and works smoothly under the normal wind speed when cyclists are riding the bike. The MPPT logic is capable of controlling the charging and discharging moments under normal and extreme wind situations.

In terms of circuit design, we designed a PCB that contained efficient circuitry for converting voltages, charging a battery and withstand unexpected situations. It is able to harvest wind power and efficiently charge a battery with the help of the MPPT algorithm.

3.2 Uncertainty

In terms of the MPPT algorithm, the accuracy for predicting the maximum power power point still is still questionable. Currently, a simple fan box is used to measure the maximum power point and fan speed with different trials. As a result, the wind speed and voltage output of anemometer may not corresponds to the point where maximum power point lies. A better testing equipment and environment should be established to provide better readings.

3.3 Ethics

Following the guidelines from the IEEE Code of Ethics, this project is developed to hold paramount the safety, health and welfare of the public.[8] All the mechanisms we designed are safe to be attached to a bicycle, to avoid any possible accidents due to a piece of our project falling off or distracting the rider.

To manage this problem, comprehensive testing on different weather conditions and the needed precautions to protect ourselves form injury are imperative to make sure that no dangerous object is manufactured and product itself is functional.

Also, some criteria in the project related the environmental aspect should be emphasized. Since the project serves as educational purpose, knowledge on environmental concern should be acquired after using the product. For this reason, all the components used are RoHS compliant and lead-freely soldered.

3.4 Future Work

- **Use more efficient anemometers to assist with MPPT.** As current hot wire anemometer is drawing too much of the power, switching to a lower power consumption anemometer should be the conducted. Specifically, lower power consumption pressure sensors should be the right way to calculate the wind speed instead. Another choice could be allowing the microcontroller to learn the perfect voltage value by itself. Specifically, microcontroller will be granted with capability to tweak the voltage and see if that value gives a higher power and to lean towards higher power direction.

- **Investigate on more efficient blades or motors in achieving the optimal theoretical power.** In the semester, 2 motors generators has been bought and tested in total, it turned out that the second generator does provide greater power. Therefore, it is possible to find a better motor that can produce greater amount of power.
- **Perform more precise MPPT measurements** In the testing procedure, the fan box was used as the main tool to get voltage and power from anemometer and maximum power point of the generator. However, measuring the anemometer and maximum power point of generator cannot be done at the same time, as the fan box is too small. Instead, we split the measurement into 2 different trials and assumed they have same wind condition, which may leads to inconsistency between the two measurements. A big wind tunnel should be used instead to create MPPT look up table.

References

[1] *"Universal Serial Bus Specification"*, Rev. 2.0, USB-IF, Apr. 2000.

[2] Y. Errami, M. Ouassaid, and M. Maaroufi, "Control of a pmsg based wind energy generation system for power maximization and grid fault conditions," *Energy Procedia*, vol. 42, pp. 220–229, 2013, Mediterranean Green Energy Forum 2013: Proceedings of an International Conference MGEF-13, ISSN: 1876-6102. DOI: https://doi.org/10.1016/j.egypro.2013.11.022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1876610213017244.

[3] J. Webprojects. ""Wind Turbine Power Calculator"." (), [Online]. Available: https://rechneronline.de/wind-power/ (visited on 05/03/2022).

[4] windynation. ""Tip Speed Ratio: How to Calculate and Apply TSR to Blade Selection"." (), [Online]. Available: https://www.windynation.com/jzv/inf/tip-speed-ratio-how-calculate-and-apply-tsr-blade-selection (visited on 05/03/2022).

[5] *"TPS63070 2-V to 16-V Buck-Boost Converter With 3.6-A Switch Current datasheet (Rev. B)"*, Rev. B, Texas Instruments, 2019. [Online]. Available: https://www.ti.com/product/TPS63070.

[6] *"SDR0805 Series - SMD Power Inductors"*, Bourns, Apr. 2018. [Online]. Available: https://www.mouser.com/datasheet/2/54/sdr0805-1391344.pdf.

[7] Tenergy. ""Tenergy 3.7V 2600mAh Li-Ion 18650 Button Top Rechargeable Battery with PCB"." (), [Online]. Available: https://www.tenergybattery.com/_files/ugd/78a53e_62683a81bf4c4080bc7b7b63b68fa766.pdf (visited on 05/03/2022).

[8] IEEE. ""IEEE Code of Ethics"." (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 02/08/2020).

Appendix A Requirements & Verification Tables

A.1 Motor Subsystem

Requirement	**Verification**
When the input socket is provided with power and no control signals are HIGH, power goes to output.	1) Apply 12V to the input with a lab power supply. 2) Verify that the output is around 12V, with a diode and V_{GS} drop.
The motor can be disconnected upon the microcontroller's request.	(Continuing the above) 3) Set `MCUO_INPUT_DISABLE` pin to HIGH (5V). 4) Verify that the output drops to around 0, and no current is drawn from lab power supply.
The motor can be forcefully slowed down upon the microcontroller's request.	(Continuing the above) 1) Put a $50mA$ current limit on the lab power supply. 2) Set `MCUO_INPUT_DISABLE` pin to LOW and set `MCUO_FORCE_DISCRG` pin HIGH. Verify that more current is being drawn from the power supply, and the supply voltage should have dropped.
The system should be able to tolerate the turbine spinning in reverse.	1) Apply -12V to the input with a lab power supply. 2) Verify that no current is being drawn from the power. 3) Verify that the output is not negative.

A.2 Input Regulator

Requirement	**Verification**
The module should be able to draw current from the input within 100ms as the input rises to 4V.	1) Connect a lab power supply to input and Oscilloscope Channel 1. Connect the positive battery terminal to Channel 2. Turn lab power supply off. Set the oscilloscope to trigger on Channel 1 rising edge @ 2V. 2) Set lab power supply to 4V and turn on. The oscilloscope should trigger. Observe the time difference between the rising of Channel 1 and Channel 2. It should not exceed 100ms.
When the input rises above 4V until 15V, the output should be 5V.	(Continuing the above) 3) Change lab power supply output voltage across the range, verify the output is 5V.
When the output is limited to 3.7-4.2V by the battery, the regulator should still be able to provide current.	(Continuing the above) 4) Connect a battery to the output, in series with an Amp meter. Verify that current is flowing into the battery, even if the Power Good signal might be off.
When the battery is charging from the input and output is disabled, the charging efficiency should be around 80%	(Continuing the above) 5) Measure the input voltage, input current, battery voltage and battery current 6) Check the difference in total input power and charging power should not exceed 20%

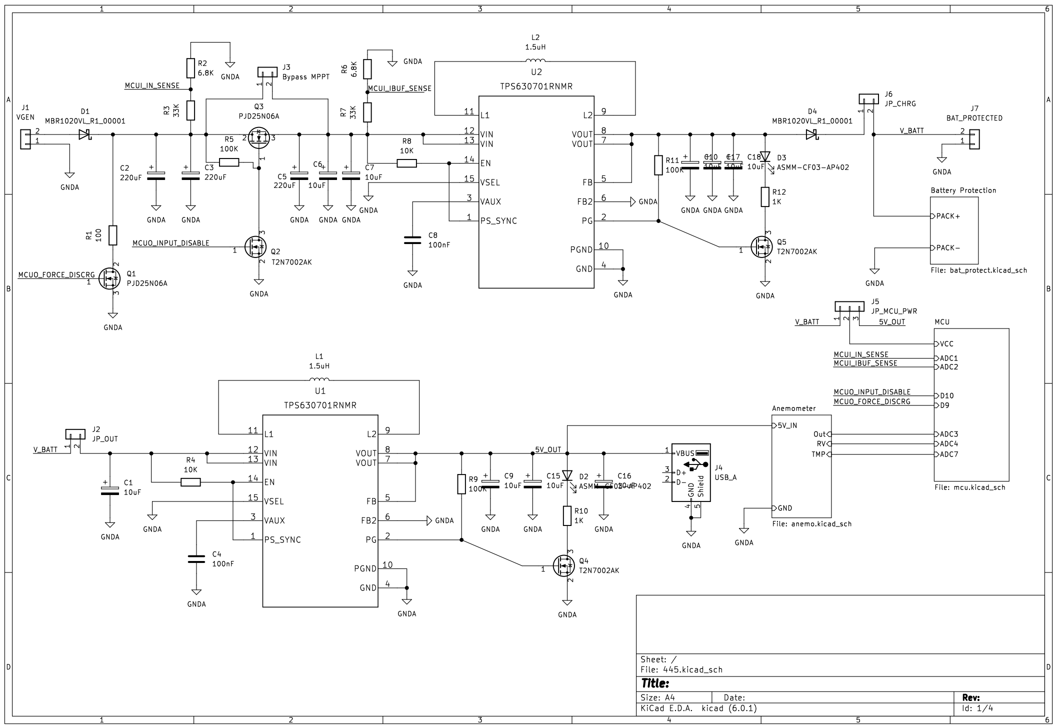
A.3 Microcontroller

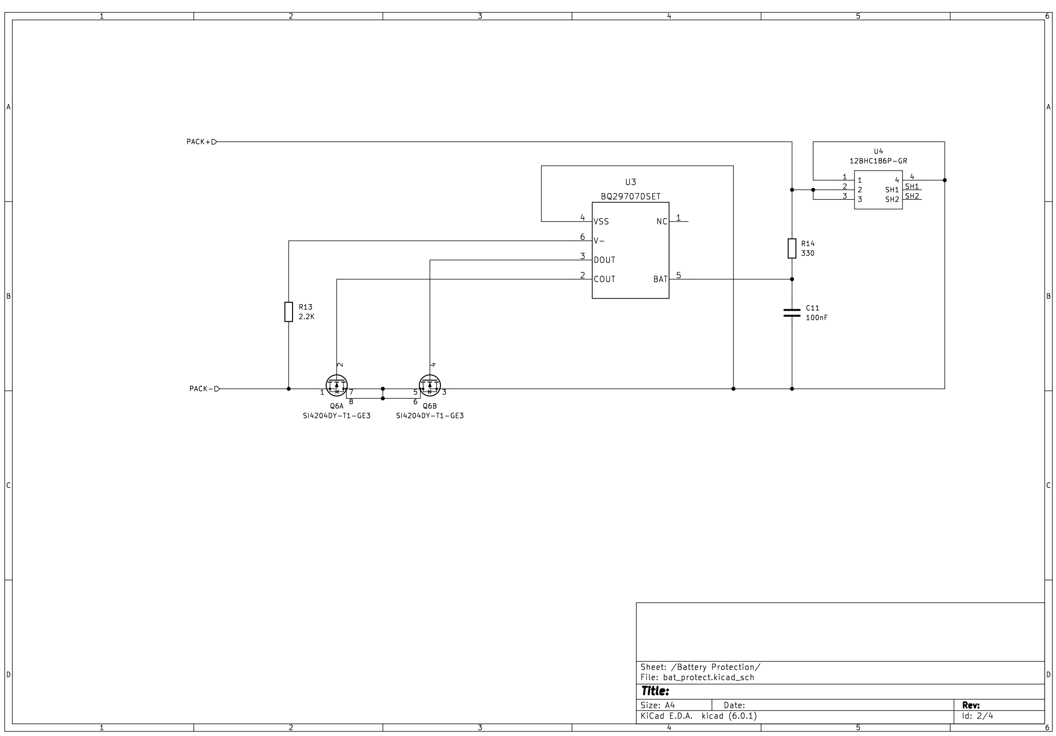
Requirement	Verification
The microcontroller should disable power input if input voltage is lower than optimal.	1) Power the microcontroller from USB programmer/UART cable. Use oscilloscope to measure D10 (MCUO_INPUT_DISABLE) signal. 2) Blow wind onto the anemometer, check the optimal voltage output on the microcontroller. 3) Using the lab power supply, provide the input with a voltage lower than the optimal voltage. Pin D10 should go high.
The microcontroller should enable power input if input voltage is higher than optimal.	(Following the above) 4) Provide the input with a voltage higher than optimal voltage. Pin D10 should go low.

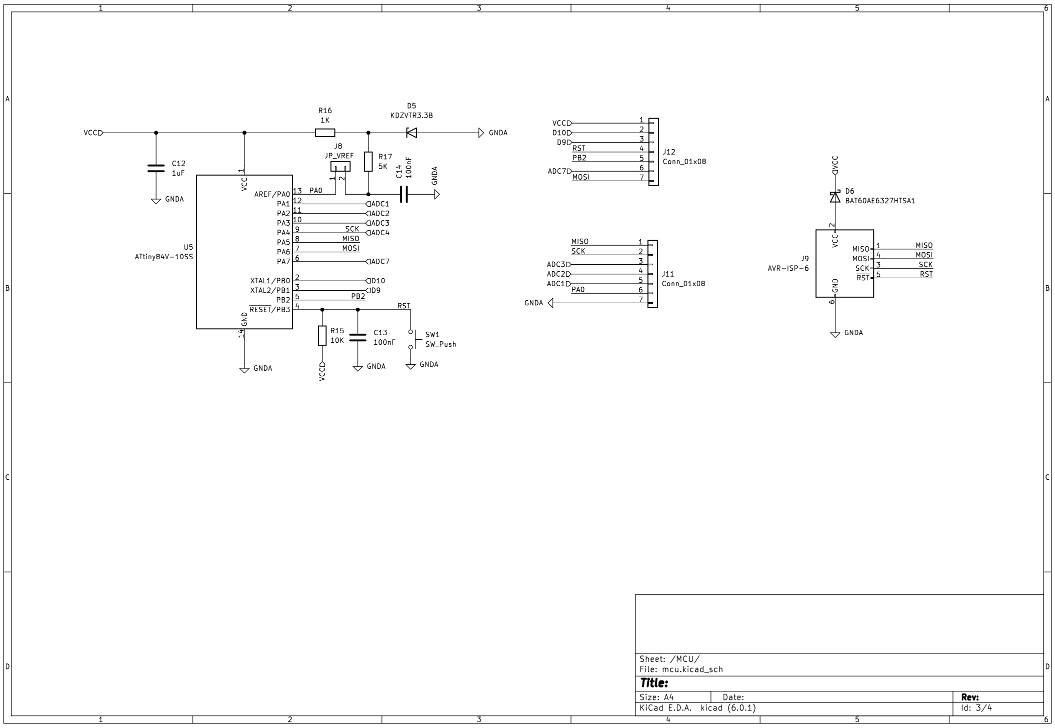
A.4 Output Regulator

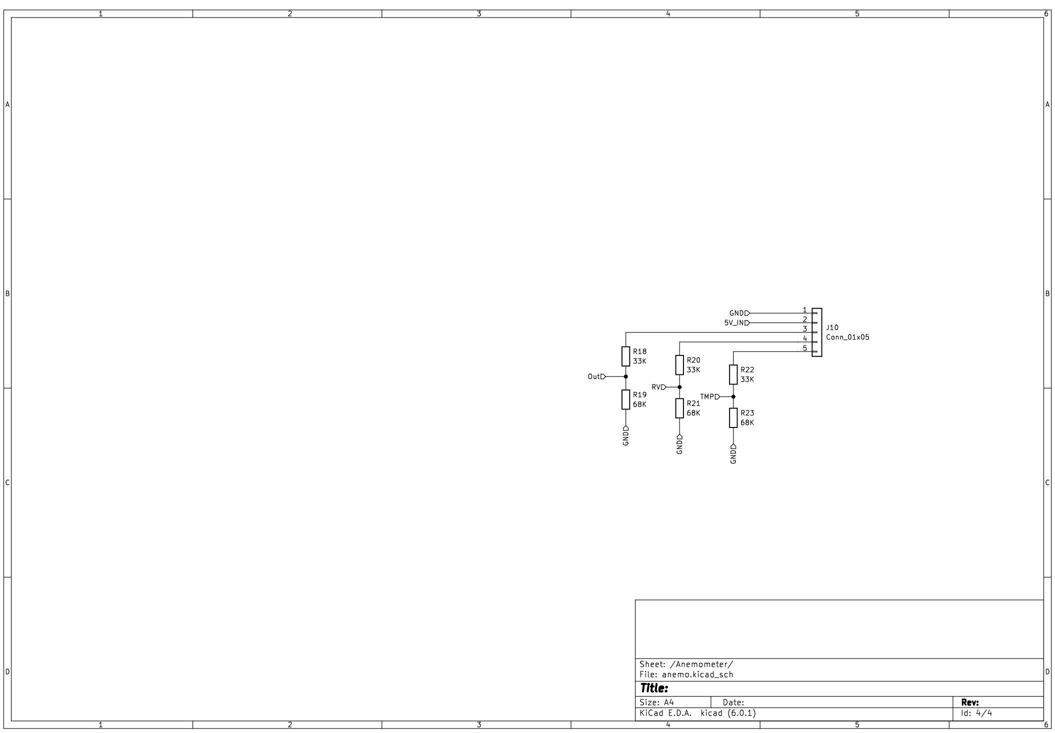
Requirement	Verification
When no load is connected, the output should be 5V.	1) Connect a battery, or a 3.9V, 2A lab power supply to the input line to simulate a battery. 2) Measure the output voltage. It should be around 5V.
The 5V line should be able to provide 500mA of current when power is available in the system.	(Continuing the above) 3) Connect the 5V output line to an electronic load and draw 500mA of current. The voltage should stay within USB specification range (4.75 – 5.25V)
The USB plugged in device should show that is charging	Connect a phone to the USB port and see the charging symbol.

Appendix B Full schematics






VCC
C12
1uF
GNDA
R16
1K
D5
KDZVTR3.3B
GNDA
J8
JP_VREF
R17
5K
C14
100nF
U5
ATtiny84V-10SS
AREF/PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
XTAL1/PB0
XTAL2/PB1
PB2
RESET/PB3
ADC1
ADC2
ADC3
ADC4
SCK
MISO
MOSI
ADC7
D10
D9
RST
R15
10K
C13
100nF
SW1
SW_Push
J12
Conn_01x08
J11
Conn_01x08
D6
BAT60AE6327HTSA1
J9
AVR-ISP-6
Sheet: /MCU/
File: mcu.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (6.0.1)
Id: 3/4


GNDD
5V_IND
J10
Conn_01x05
R18
33K
R19
68K
Out
R20
33K
R21
68K
RVD
R22
33K
R23
68K
TMPD
Sheet: /Anemometer/
File: anemo.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (6.0.1)
Id: 4/4

Appendix C   Regulator Testing Data

C.1   Input Regulator

Table 5: Input Regulator Testing Data

Input V(V)	Input I(A)	Input P(W)	Output V(V)	Output I(A)	Output P(W)	Efficiency
3.004	0.006	0.0180	5.0319	0	0	0
3.003	0.03	0.0900	4.8768	0.0075	0.0365	0.4059
3	0.077	0.231	4.8498	0.0289	0.1401	0.6067
3.001	0.123	0.3691	4.8378	0.0494	0.2389	0.6474
3.001	0.236	0.7082	4.8097	0.099	0.4761	0.6723
3.001	0.284	0.8522	4.793	0.1192	0.5713	0.6703
3	0.332	0.996	4.7691	0.1386	0.6609	0.6636
3.001	0.383	1.1493	4.7529	0.1592	0.7566	0.6583
3.001	0.441	1.3234	4.7481	0.1793	0.8513	0.6432
3.001	0.505	1.5155	4.7445	0.1998	0.9479	0.6255
3	0.55	1.65	4.7419	0.22	1.0432	0.6322
3	0.604	1.812	4.7394	0.2393	1.1341	0.6259
3	0.68	2.04	4.7373	0.2597	1.2302	0.6030
3	0.629	1.887	4.748	0.2789	1.3242	0.7017
3	0.651	1.953	4.7365	0.2994	1.4181	0.7261
5.999	0.002	0.0119	5.0393	0	0	0
5.999	0.013	0.0779	4.8747	0.0075	0.0365	0.4687
5.999	0.034	0.2039	4.847	0.0289	0.1400	0.6867
5.999	0.054	0.3239	4.8353	0.0495	0.2393	0.7388
5.999	0.103	0.6178	4.8196	0.0991	0.4776	0.7729
5.999	0.124	0.7438	4.8135	0.1193	0.5742	0.7719
5.999	0.14	0.8398	4.8054	0.1386	0.6660	0.7930
5.999	0.161	0.9658	4.8003	0.1591	0.7637	0.7907
5.999	0.182	1.0918	4.7923	0.1794	0.8597	0.7874

Table 5: Input Regulator Testing Data (Continued)

Input V(V)	Input I(A)	Input P(W)	Output V(V)	Output I(A)	Output P(W)	Efficiency
5.999	0.199	1.1938	4.7727	0.1997	0.9531	0.7983
5.999	0.22	1.3197	4.7465	0.2199	1.0437	0.7908
5.999	0.24	1.4397	4.7369	0.2393	1.1335	0.7873
5.999	0.262	1.5717	4.7335	0.2596	1.2288	0.7818
5.999	0.283	1.6977	4.7295	0.2789	1.3190	0.7769
5.999	0.291	1.7457	4.7471	0.2993	1.4208	0.8138
8.998	0.002	0.0179	5.0463	0	0	0
8.998	0.009	0.0809	4.8683	0.0074	0.0360	0.4448
8.998	0.022	0.1979	4.8401	0.029	0.1403	0.7090
8.998	0.036	0.3239	4.8252	0.0494	0.2383	0.7358
8.998	0.068	0.6118	4.8069	0.0991	0.4763	0.7785
8.998	0.081	0.7288	4.8024	0.1193	0.5729	0.7860
8.998	0.094	0.8458	4.7975	0.1387	0.6654	0.7867
8.998	0.107	0.9627	4.7911	0.159	0.7617	0.7912
8.998	0.12	1.0797	4.7771	0.1794	0.8570	0.7937
8.998	0.132	1.1877	4.75	0.1996	0.9481	0.7982
8.998	0.146	1.3137	4.724	0.22	1.0392	0.7911
8.999	0.158	1.4218	4.724	0.24	1.1337	0.7973
8.999	0.172	1.5478	4.725	0.26	1.2285	0.7936
8.999	0.184	1.6558	4.726	0.28	1.3232	0.7991
8.999	0.197	1.7728	4.728	0.3	1.4184	0.8000
12	0.001	0.012	5.0831	0	0	0
12	0.007	0.084	4.8689	0.0075	0.0365	0.4347
12	0.017	0.204	4.8457	0.0293	0.1419	0.6959
12	0.028	0.336	4.8343	0.0496	0.2397	0.7136
12	0.053	0.636	4.8193	0.0994	0.4790	0.7532

Table 5: Input Regulator Testing Data (Continued)

Input V(V)	Input I(A)	Input P(W)	Output V(V)	Output I(A)	Output P(W)	Efficiency
12.001	0.063	0.7560	4.8164	0.119	0.5731	0.7580
12	0.072	0.864	4.814	0.1386	0.6672	0.7722
12	0.083	0.996	4.8115	0.1584	0.7621	0.7652
12	0.093	1.116	4.8082	0.1799	0.8649	0.7750
12	0.103	1.236	4.8051	0.1995	0.9586	0.7755
12	0.113	1.356	4.8016	0.22	1.0563	0.7790
12	0.123	1.476	4.7972	0.2386	1.1446	0.7754
12	0.132	1.584	4.7903	0.2594	1.2426	0.7844
12	0.142	1.704	4.7807	0.279	1.3338	0.7827
12	0.151	1.812	4.7573	0.2994	1.4243	0.7860
13.999	0.001	0.0139	5.1028	0	0	0
13.999	0.006	0.0839	4.8719	0.0072	0.0350	0.4176
13.999	0.015	0.2099	4.8487	0.029	0.1406	0.6696
13.999	0.024	0.3359	4.8368	0.0492	0.2379	0.7082
13.999	0.046	0.6439	4.8234	0.0986	0.4755	0.7385
13.999	0.055	0.7699	4.8208	0.119	0.5736	0.7450
13.999	0.063	0.8819	4.8192	0.1387	0.6684	0.7579
13.999	0.072	1.0079	4.8179	0.1592	0.7670	0.7609
13.999	0.081	1.1339	4.8162	0.1797	0.8654	0.7632
13.999	0.09	1.2599	4.814	0.1994	0.9599	0.7618
13.999	0.098	1.3719	4.8108	0.2194	1.0554	0.7693
13.998	0.107	1.4977	4.8084	0.2394	1.1511	0.7685
13.999	0.116	1.6238	4.805	0.2592	1.2454	0.7669
13.998	0.124	1.7357	4.8023	0.2793	1.3412	0.7727
13.998	0.133	1.8617	4.7977	0.2994	1.4364	0.7715
16	0.001	0.016	5.1158	0	0	0

Table 5: Input Regulator Testing Data (Continued)

Input V(V)	Input I(A)	Input P(W)	Output V(V)	Output I(A)	Output P(W)	Efficiency
16	0.005	0.08	4.8736	0.0077	0.0375	0.4690
16	0.014	0.224	4.8513	0.0286	0.1387	0.6194
16	0.022	0.352	4.8401	0.0491	0.2376	0.6751
16	0.041	0.656	4.8259	0.0986	0.4758	0.7253
16	0.049	0.784	4.8243	0.1195	0.5765	0.7353
16	0.056	0.896	4.823	0.1382	0.6665	0.7439
16	0.064	1.024	4.8225	0.1592	0.7677	0.7497
16	0.072	1.152	4.8218	0.1795	0.8655	0.7513
16	0.08	1.28	4.8206	0.1993	0.9607	0.7505
16	0.088	1.408	4.819	0.2203	1.0616	0.7539
16	0.095	1.52	4.8173	0.2396	1.1542	0.7593
16	0.103	1.648	4.815	0.2601	1.2523	0.7599
16	0.111	1.776	4.8128	0.2788	1.3418	0.7555
16	0.118	1.888	4.8102	0.2992	1.4392	0.7622

C.2 Output Regulator

Table 6: Output Regulator Testing Data

Input V(V)	Input I(A)	Input P(W)	Output V(V)	Output I(A)	Output P(W)	Efficiency
3.702	0.004	0.0148	5.0445	0	0	0
3.702	0.023	0.0851	5.048	0.0071	0.0358	0.4209
3.702	0.061	0.2258	5.0438	0.029	0.1462	0.6477
3.702	0.098	0.3627	5.0368	0.0494	0.2488	0.6858
3.702	0.184	0.6811	5.0263	0.0995	0.5001	0.7342
3.702	0.269	0.9958	5.0116	0.1492	0.7477	0.7508
3.702	0.355	1.3142	4.975	0.1995	0.9925	0.7552
3.702	0.444	1.6436	4.9628	0.2493	1.2372	0.7527

Table 6: Output Regulator Testing Data (Continued)

Input V(V)	Input I(A)	Input P(W)	Output V(V)	Output I(A)	Output P(W)	Efficiency
3.702	0.529	1.9583	4.9523	0.2993	1.4822	0.7568
3.702	0.642	2.3766	4.9333	0.3496	1.7246	0.7256
3.702	0.602	2.2286	4.9379	0.3987	1.9687	0.8833
3.702	0.689	2.5506	4.9296	0.4488	2.2124	0.8673
3.702	0.775	2.8690	4.9227	0.5004	2.4633	0.8585
3.702	0.861	3.1874	4.9157	0.5502	2.7046	0.8485
3.702	0.947	3.5057	4.9091	0.5999	2.9449	0.8400
3.702	1.035	3.8315	4.9028	0.6497	3.1853	0.8313
3.702	1.123	4.1573	4.8962	0.7004	3.4292	0.8248
3.702	1.217	4.5053	4.89	0.7501	3.6679	0.8141
3.702	1.311	4.8533	4.8839	0.8	3.9071	0.8050
3.702	1.408	5.2124	4.8774	0.8496	4.1438	0.7949
3.702	1.51	5.5900	4.8714	0.8997	4.3827	0.7840
3.702	1.613	5.9713	4.8657	0.9504	4.6243	0.7744
3.702	1.72	6.3674	4.8604	0.9995	4.8579	0.7629
3.802	0.004	0.0152	5.0441	0	0	0
3.802	0.023	0.0874	5.0477	0.0071	0.0358	0.4098
3.802	0.06	0.2281	5.0439	0.029	0.1462	0.6412
3.802	0.095	0.3611	5.037	0.0493	0.2483	0.6875
3.802	0.178	0.6767	5.0269	0.0994	0.4996	0.7383
3.802	0.26	0.9885	5.0135	0.149	0.7470	0.7556
3.802	0.343	1.3040	4.9763	0.1993	0.9917	0.7605
3.802	0.429	1.6310	4.9589	0.2498	1.2387	0.7594
3.802	0.511	1.9428	4.9532	0.2992	1.4819	0.7628
3.802	0.61	2.3192	4.9331	0.3497	1.7251	0.7438
3.802	0.585	2.2241	4.9225	0.3992	1.9650	0.8835

Table 6: Output Regulator Testing Data (Continued)

Input V(V)	Input I(A)	Input P(W)	Output V(V)	Output I(A)	Output P(W)	Efficiency
3.802	0.669	2.5435	4.9296	0.4493	2.2148	0.8707
3.802	0.75	2.8515	4.923	0.5006	2.4644	0.8642
3.802	0.833	3.1670	4.9156	0.55	2.7035	0.8536
3.802	0.915	3.4788	4.9091	0.6003	2.9469	0.8471
3.802	0.998	3.7943	4.9026	0.6495	3.1842	0.8391
3.802	1.083	4.1175	4.8963	0.7001	3.4278	0.8325
3.802	1.171	4.4521	4.8895	0.75	3.6671	0.8236
3.802	1.258	4.7829	4.8835	0.8	3.9068	0.8168
3.802	1.352	5.1403	4.8772	0.85	4.1456	0.8064
3.802	1.447	5.5014	4.871	0.9002	4.3848	0.7970
3.802	1.545	5.8740	4.8646	0.9505	4.6238	0.7871
3.802	1.643	6.2466	4.8591	1.0001	4.8595	0.7779
4.002	0.004	0.0160	5.0441	0	0	0
4.002	0.022	0.0880	5.0473	0.0077	0.0388	0.4414
4.002	0.056	0.2241	5.0462	0.0291	0.1468	0.6552
4.002	0.09	0.3601	5.0389	0.0499	0.2514	0.6980
4.002	0.169	0.6763	5.0298	0.0986	0.4959	0.7332
4.002	0.245	0.9804	5.0172	0.1497	0.7510	0.7660
4.002	0.323	1.2926	4.9933	0.1995	0.9961	0.7706
4.002	0.403	1.6128	4.955	0.25	1.2387	0.7680
4.002	0.479	1.9169	4.946	0.2993	1.4803	0.7722
4.002	0.563	2.2531	4.9321	0.3496	1.7242	0.7652
4.002	0.666	2.6653	4.9258	0.3986	1.9634	0.7366
4.002	0.622	2.4892	4.9299	0.4496	2.2164	0.8904
4.002	0.702	2.8094	4.9225	0.5001	2.4617	0.8762
4.002	0.779	3.1175	4.9158	0.5502	2.7046	0.8675

Table 6: Output Regulator Testing Data (Continued)

Input V(V)	Input I(A)	Input P(W)	Output V(V)	Output I(A)	Output P(W)	Efficiency
4.002	0.855	3.4217	4.9088	0.6005	2.9477	0.8614
4.002	0.932	3.7298	4.9025	0.6495	3.1841	0.8536
4.002	1.01	4.0420	4.8959	0.6998	3.4261	0.8476
4.002	1.09	4.3621	4.8895	0.7503	3.6685	0.8409
4.002	1.171	4.6863	4.8826	0.7999	3.9055	0.8333
4.002	1.255	5.0225	4.8765	0.8499	4.1445	0.8251
4.002	1.34	5.3626	4.8697	0.9005	4.3851	0.8177
4.002	1.428	5.7148	4.8635	0.9503	4.6217	0.8087
4.002	1.517	6.0710	4.8574	0.9995	4.8549	0.7996
4.202	0.004	0.0168	5.0438	0	0	0
4.202	0.02	0.0840	5.0475	0.0073	0.0368	0.4384
4.202	0.053	0.2227	5.048	0.0291	0.1468	0.6595
4.202	0.085	0.3571	5.0412	0.0487	0.2455	0.6873
4.202	0.16	0.6723	5.0321	0.0995	0.5006	0.7447
4.202	0.234	0.9832	5.0202	0.1495	0.7505	0.7632
4.202	0.306	1.2858	5.0025	0.1992	0.9964	0.7749
4.202	0.379	1.5925	4.9469	0.2501	1.2372	0.7768
4.202	0.453	1.9035	4.9414	0.2991	1.4779	0.7764
4.202	0.531	2.2312	4.9309	0.35	1.7258	0.7734
4.202	0.614	2.5800	4.9253	0.3995	1.9676	0.7626
4.202	0.587	2.4665	4.9304	0.4494	2.2157	0.8982
4.202	0.679	2.8531	4.9205	0.4996	2.4582	0.8616
4.202	0.748	3.1430	4.9144	0.5505	2.7053	0.8607
4.202	0.823	3.4582	4.907	0.6003	2.9456	0.8517
4.202	0.894	3.7565	4.9004	0.6494	3.1823	0.8471
4.202	0.973	4.0885	4.8933	0.7003	3.4267	0.8381

Table 6: Output Regulator Testing Data (Continued)

Input V(V)	Input I(A)	Input P(W)	Output V(V)	Output I(A)	Output P(W)	Efficiency
4.202	1.049	4.4078	4.8863	0.7505	3.6671	0.8319
4.202	1.123	4.7188	4.8803	0.7995	3.9017	0.8268
4.202	1.206	5.0676	4.8733	0.8499	4.1418	0.8173
4.202	1.278	5.3701	4.8677	0.9007	4.3843	0.8164
4.202	1.365	5.7357	4.8608	0.9504	4.6197	0.8054
4.202	1.442	6.0592	4.855	0.9995	4.8525	0.8008

Appendix D Microcontroller code

```
//#define MCU_IN_DISABLE_PIN A0
//#define analogPinForRV A3
//#define analogPinForTMP A4
//#define analogPinForOUT A5
//#define MCUI_IN_SENSE A7

#include <SoftwareSerial.h>
#include "tinysnore.h"

#define MCU_IN_DISABLE_PIN 10
#define MCUI_IBUF_SENSE A2
#define MCUO_FORCE_DISCRG 9
#define analogPinForRV A4
#define analogPinForTMP A7
#define analogPinForOUT A3
#define MCUI_IN_SENSE A1

#define DEBUG
#define USE_ANEMOMETER

#define rxPin 6
#define txPin 5
// analogPinForOrigVolt is the MCU_IN_SENSE

const float zeroWindAdjustment = 0.44;

float R2 = 6800.0;
float R3 = 33000.0;
float best_voltage;
float orig_volt;
float orig_IBUF_volt;
uint32_t lastMillis = 0;
int16_t TMP_Therm_ADunits; // temp termistor value from wind sensor
int16_t RV_Wind_ADunits; // RV output from wind sensor
float RV_Wind_Volts;
// int TempCtimes100;
double zeroWind_ADunits;
double zeroWind_volts;
float WindSpeed_MPH;
float OUT;
int16_t OUT_unit;

#ifdef DEBUG
SoftwareSerial mySerial = SoftwareSerial(rxPin, txPin);
#endif
```

```
void setup()
{
   // put your setup code here, to run once:
#ifdef DEBUG
   mySerial.begin(57600);
   mySerial.println("B");
#endif
   pinMode(MCUI_IN_SENSE, INPUT);
   pinMode(MCUI_IBUF_SENSE, INPUT);
   pinMode(analogPinForRV, INPUT);
   pinMode(analogPinForTMP, INPUT);
   pinMode(analogPinForOUT, INPUT);
   pinMode(MCU_IN_DISABLE_PIN, OUTPUT);
   pinMode(MCUO_FORCE_DISCRG, OUTPUT);

   pinMode(rxPin, INPUT);
   pinMode(txPin, OUTPUT);
}

void loop()
{
#ifdef DEBUG
   mySerial.println("S:");
#endif

   int16_t in_sense_raw = analogRead(MCUI_IN_SENSE);
   int16_t mcui_ibuf_sense_raw = analogRead(MCUI_IBUF_SENSE);
   orig_volt = in_sense_raw * 5.0 / 1023.0 * (R2 + R3) / R2;
   orig_IBUF_volt = mcui_ibuf_sense_raw * 5.0 / 1023.0 * (R2 + R3) /
      R2;

#ifdef USE_ANEMOMETER
   // Units for anemometer
   TMP_Therm_ADunits = analogRead(analogPinForTMP);
   RV_Wind_ADunits = analogRead(analogPinForRV);
   OUT_unit = analogRead(analogPinForOUT);

#ifdef DEBUG
   mySerial.print("I_V_R:");
   mySerial.println(in_sense_raw);
#endif
   RV_Wind_Volts = (RV_Wind_ADunits * 0.0048828125);
   OUT = (OUT_unit * 0.0048828125);

   // Calculating windspeed
```

```
    // TempCtimes100 = (0.005 *((float)TMP_Therm_ADunits *
       (float)TMP_Therm_ADunits)) - (16.862 * (float)TMP_Therm_ADunits)
       + 9075.4;
    zeroWind_ADunits = -0.0006 * ((float)TMP_Therm_ADunits *
       (float)TMP_Therm_ADunits) + 1.0727 * (float)TMP_Therm_ADunits +
       47.172; // 13.0C 553 482.39
    zeroWind_volts = (zeroWind_ADunits * 0.0048828125) -
       zeroWindAdjustment;
    WindSpeed_MPH = pow(((RV_Wind_Volts - zeroWind_volts) / .2300),
       2.7265);

    if (OUT > 1.775)
    {
       best_voltage = 2.3481 * OUT + 8.0752;
    }
    else
    {
       best_voltage = 8.9693 * OUT + 1.775;
    }
#else
    best_voltage = 12.0;
#endif

    if(best_voltage <= 6) {
     best_voltage = 6.0;
    }
    // MCU logic
    if (orig_volt >= 16)
    {
       digitalWrite(MCUO_FORCE_DISCRG, HIGH);
       digitalWrite(MCU_IN_DISABLE_PIN, LOW);
#ifdef DEBUG
       mySerial.println("C-1");
#endif
    }
    else
    {
       if (orig_IBUF_volt > orig_volt)
       {
#ifdef DEBUG
          mySerial.println("C-2");
#endif
          digitalWrite(MCUO_FORCE_DISCRG, LOW);
          digitalWrite(MCU_IN_DISABLE_PIN, LOW);
       }
       else
       {
```

```
        if (orig_volt > best_voltage)
        {
#ifdef DEBUG
            mySerial.println("C-3");
#endif
            digitalWrite(MCUO_FORCE_DISCRG, LOW);
            digitalWrite(MCU_IN_DISABLE_PIN, LOW);
        }
        else
        {
#ifdef DEBUG
            mySerial.println("C-4");
#endif
            digitalWrite(MCUO_FORCE_DISCRG, LOW);
            digitalWrite(MCU_IN_DISABLE_PIN, HIGH);
        }
      }
   }

#ifdef DEBUG
   String prompt = "Anemo:";
   prompt.concat(OUT);
   mySerial.println(prompt);

   prompt = "Best V:";
   prompt.concat(best_voltage);
   mySerial.println(prompt);

   prompt = "In V:";
   prompt.concat(orig_volt);
   mySerial.println(prompt);
#endif

   snore(1000);
}
```

Appendix E MPPT Measurement Data

Table 7 shows testing done at a measured wind speed of $10.2134MPH$, Anemometer output voltage is $2.07534V$

Table 7: Fan speed 2, Distance 5cm

Voltage (V)	Current (A)	Power (W)	Calculated power
9	0.017	0.157	0.153
9.5	0.018	0.175	0.171
10	0.0182	0.189	0.182
10.5	0.0185	0.2	0.19425
11	0.0215	0.235	0.2365
11.5	0.0195	0.225	0.22425
12	0.0187	0.223	0.2244
12.5	0.0167	0.21	0.20875
13	0.0153	0.193	0.1989
13.5	0.0125	0.175	0.16875

Table 8 shows testing done at a measured wind speed of $7.7439MPH$, Anemometer output voltage is $1.99023V$

Table 9 shows testing done at a measured wind speed of $5.8437MPH$, Anemometer output voltage is $1.7737V$

Table 10 shows testing done at a measured wind speed of $3.92417MPH$, Anemometer output voltage is $1.55833V$

Table 11 shows testing done at a measured wind speed of $10.835MPH$, Anemometer output voltage is $2.110833V$

Table 12 shows testing done at a measured wind speed of $11.429MPH$, Anemometer output voltage is $2.151V$

Table 13 shows testing done at a measured wind speed of $7.485MPH$, Anemometer output voltage is $1.990833V$

Table 14 shows testing done at a measured wind speed of $10.88083MPH$, Anemometer output voltage is $2.035V$

Table 15 shows the final MPPT result.

Table 8: Fan speed 2, Distance 10cm

Voltage (V)	Current (A)	Power (W)	Calculated power
9	0.0113	0.106	0.1017
9.5	0.0115	0.11	0.10925
10	0.012	0.117	0.12
10.5	0.0102	0.108	0.1071
11	0.0179	0.175	0.1969
11.5	0.0144	0.177	0.1656
12	0.0128	0.164	0.1536
12.5	0.0119	0.144	0.14875
13	0.0101	0.128	0.1313
13.5	0.0087	0.114	0.11745

Table 9: Fan speed 2, Distance 15cm

Voltage (V)	Current (A)	Power (W)	Calculated power
9	0.0101	0.087	0.0909
9.5	0.0095	0.087	0.09025
10	0.0086	0.085	0.086
10.5	0.079	0.081	0.8295
11	0.0067	0.079	0.0737
11.5	0.0057	0.064	0.06555
12	0.0039	0.045	0.0468
12.5	0.0028	0.03	0.035
13	0.001	0.013	0.013

Table 10: Fan speed 2, Distance 20cm

Voltage (V)	Current (A)	Power (W)	Calculated power
9	0.0124	0.105	0.1116
9.5	0.0113	0.108	0.10735
10	0.011	0.105	0.11
10.5	0.0098	0.102	0.1029
11	0.085	0.096	0.935
11.5	0.0063	0.078	0.07245
12	0.0049	0.068	0.0588
12.5	0.0044	0.053	0.055
13	0.0031	0.039	0.0403
13.5	0.0012	0.014	0.0162

Table 11: Fan speed 3, Distance 5cm

Voltage (V)	Current (A)	Power (W)	Calculated power
9	0.0374	0.332	0.3366
9.5	0.0371	0.364	0.35245
10	0.0361	0.363	0.361
10.5	0.0376	0.399	0.3948
11	0.0343	0.373	0.3773
11.5	0.0331	0.378	0.38065
12	0.0317	0.389	0.3804
12.5	0.0302	0.378	0.3775
13	0.0284	0.368	0.3692
13.5	0.0277	0.372	0.37395

Table 12: Fan speed 3, Distance 10cm

Voltage (V)	Current (A)	Power (W)	Calculated power
9	0.0371	0.33	0.3339
9.5	0.0355	0.338	0.33725
10	0.0379	0.371	0.379
10.5	0.0374	0.397	0.3927
11	0.0336	0.37	0.3696
11.5	0.0339	0.387	0.38985
12	0.0307	0.377	0.3684
12.5	0.0303	0.371	0.37875
13	0.0266	0.357	0.3458
13.5	0.025	0.339	0.3375

Table 13: Fan speed 3, Distance 15cm

Voltage (V)	Current (A)	Power (W)	Calculated power
9	0.0385	0.353	0.3465
9.5	0.0387	0.375	0.36765
10	0.0374	0.375	0.374
10.5	0.0373	0.392	0.39165
11	0.0329	0.363	0.3619
11.5	0.0299	0.34	0.34385
12	0.0276	0.337	0.3312
12.5	0.027	0.342	0.3375
13	0.0248	0.338	0.3224
13.5	0.0225	0.315	0.30375

Table 14: Fan speed 3, Distance 20cm

Voltage (V)	Current (A)	Power (W)	Calculated power
9	0.0354	0.303	0.3186
9.5	0.0339	0.326	0.32205
10	0.0335	0.33	0.335
10.5	0.0329	0.345	0.34545
11	0.0332	0.372	0.3652
11.5	0.0309	0.355	0.35535
12	0.029	0.347	0.348
12.5	0.0283	0.351	0.35375
13	0.0259	0.336	0.3367
13.5	0.0242	0.32	0.3267

Table 15: MPPT Final Result

Anemometer Out (V)	Best Generator voltage (V)
1.55833	9.5
1.7737	9.5
1.99023	11.5
1.990833	10.5
2.035	11
2.07534	11
2.110833	12
2.151	11.5