

California State University, Fresno
Lyles College of Engineering
Electrical and Computer Engineering Department

FINAL PROJECT PROPOSAL

Project Title: Energy Harvesting for Wireless Sensor Networks

Course Title: ECE186B – Senior Design II

Date Submitted: May 10, 2019

Prepared By:

Luis Miguel Ortiga, Jasmine Lopez, Jared Sarajian

Faculty Advisor:

Dr. Na, Dr. Wang

INSTRUCTOR SECTION

Comments: _____

Final Grade: Team Member 1: _____

Team Member 2: _____

Team Member 3: _____

TABLE OF CONTENTS

Section	Page
1. Abstract	2
2. Theoretical Background	2
3. Design Overview	4
3.1 List of Equipment	4
3.2 Solar	5
3.3 Power Management Design	9
3.3.1 Power Stage	9
3.3.2 Energy Storage	10
3.3.3 Control Design	13
3.3.4 Hardware Implementation	16
3.4 Simulation	19
3.4.1 Power Stage Evaluation	19
3.4.2 Compensation Results	21
3.5 Wireless Sensor Network	24
3.5.1 Wireless Sensor Mote Choice	24
3.5.2 Wireless Sensor Mote Breakdown	25
3.5.3 Contiki vs. TinyOS	26
3.5.4 Programming the TelosB	27
3.5.5 Using the Cooja Network Simulator	30
3.5.6 LPM and Interrupt Processing	31
3.5.7 Types of Casting	32
3.5.8 Multi-Hop Future Design Plan	33
3.5.9 Design with Energy Harvesting	34
3.6 IEEE Standards	34
4. End Product	35
4.1 Results	35
4.2 Proposal for Future Work	38
4.3 Budget	38
5. Logistics	39
5.1 Division of Labor	39
5.2 Work Completed	39
6. Statements about Team Members	40
7. References	41
9. Appendix A	43
10. Appendix B	44
11. Appendix C	45

1. ABSTRACT

The objective of this project was to create a wireless sensor network (WSN) that runs off of solar energy. These sensors could theoretically be placed in remote areas to gather and transmit data back to some central server, allowing the collection of data about the surrounding environment such as air pollution, humidity and temperature. The individual sensor will rely on a lithium-ion, which itself is charged in a controlled manner using a microcontroller, a two switch buck-boost DC/DC converter and a monocrystalline solar panel. This allows them to operate autonomously without the need to be connected to the power grid.

2. THEORETICAL BACKGROUND

Energy harvesting systems must be designed with careful consideration as the amount of energy being harvested can often be miniscule and/or irregular. If the system is to operate continuously, then the system must consume less energy than is available in its immediate surroundings. It must also have a means of storing excess energy that isn't used immediately as well as a means for regulating the power delivered to the load.

The sensor mote will only need to sample environmental data on an interval rather than continuously sampling and transmitting. This will extend the charge of the battery, prolonging its total life-span.

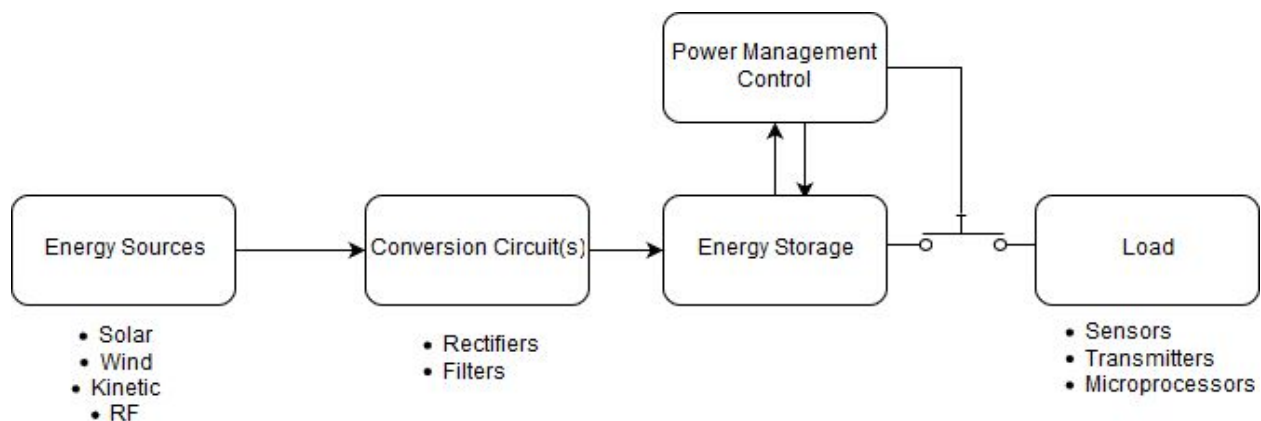


Figure 2.1: General block diagram of an energy harvesting system

Solar power was chosen as the energy source as it provides the best available power density at an average value of 1000W/m^2 . Though the theoretical maximum efficiency of a solar cell is approximately 33.7%, its affordability and practicality makes it one of the most popular choices for renewable energy harvesting.

The efficiency of the panel can be boosted by impedance matching its load in order to maximize the power delivered. This method of control falls under the class of algorithms known as maximum power point tracking (MPPT). As the name implies, the tracking algorithms continuously monitor the power delivered by the panel to a load and dynamically changes the panels operating voltage to maximize this value, assuming it doesn't exceed the loads limitations. Two of the most popular MPPT algorithms are Perturb & Observe (P&O) and Incremental Conductance (IC) which offer good performance considering their relatively simple design. The algorithms can be implemented via hardware but are normally implemented digitally using microcontrollers as they require precise numerical analysis of the voltage and current levels in the system.

The power conversion circuit will be a DC/DC two-switch buck/boost converter which will regulate the charge voltage and current supplied to the battery, implementing a constant current/constant voltage charging scheme (CC/CV). Once fully charged, the battery will provide power to the mote. This requires a feedback control system which must be able to regulate both current and voltage at or below a $\pm 1\%$ error.

The second half of this project is the wireless sensor network design. The WSN will be simulated in software to observe how the individual motes interact under varying configurations. This could be a simple 3-node network that measures ambient temperature, light and humidity (included sensor units in the CM5000 mote that will be used). The idea is to get this infrastructure working on a small scale and then scale up. If the underlying lay-out of the network nodes is configured in an efficient way they the design could be scaled to cover larger areas. The sensor nodes can be connected via various methodologies, but the most basic one is called the Star Method. It consists of a sensor mote connected out to each of the individual nodes. The sensor mote is also known as the gateway or base unit. This is the crux of the connection between the physical sensor nodes and the cloud connection. Every node can talk back and forth with the sensor mote, but they cannot laterally talk to each other. That feature is definitely sought after for certain applications, but for right now ours does not require that functionality.

Once the WSN is set up and working, the focus will be shifted to the design of the connection with the cloud. A cloud connection can be established via an internet connection. This will in turn be connected to what will likely be a 3rd party cloud service provider. Of course there will also need to be some local database as well as a cloud database just in case the data is lost during transfer. The sensor nodes and sensor mote have very little storage capacity so the local storage and cloud storage are absolutely necessary.

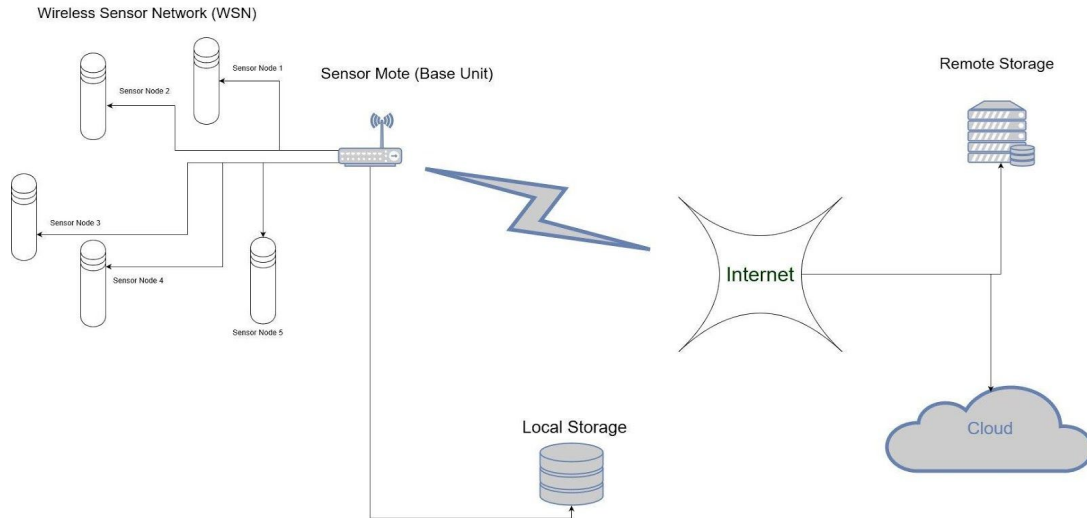


Figure 2.2: Wireless Sensor Network Operation

3. DESIGN OVERVIEW

3.1 List of Equipment

- Sensor Mote (3 at the very least)
- Local Data Storage Device
- Various Software (VMWare, Contiki OS, Matlab, Psim, MPLAB X IDE)
- Solar Panels (6V, 3.4W)
- PIC16F1778 microcontroller
- Capacitors (820 μ F, 1000 μ F)
- Inductor (100 μ F)
- MBR1535CTG Schottky diodes
- MAX4172 High-Side Current Sensors
- 19TQ015 Schottky Diodes (2)
- IRFZ14N MOSFETS
- FOD3180 Optocoupler MOSFET Gate Drivers (2)
- LT6005 1 μ A Op Amp (Quad)
- TSSOP/SSOP to DIP Adapters
- 1100mAh Li-Ion battery
- Stripboard
- Tin Lead Solder
- Flux
- Resistors
- Breadboard

3.2. Solar

There are many possible sources available for energy harvesting. For this project, the main focus was solar. Solar took priority because of its practicality in the Fresno environment. There are indoor and outdoor options available; however, indoor solar from fluorescent lighting provides much less power. Considering the sensors were planned to be placed outdoors, it was likely the sensor would benefit and receive outdoor solar. It is important to find a solar panel that is appropriately sized for this network's operation because the load will require a specific voltage and current. Since the sensor mote and the battery were estimated to draw approximately 2-3 W at maximum, a solar panel capable of providing 3.4 W was chosen. The equivalent network for a solar panel is seen in Figure 3.1 [5]. Higher efficiency solar panels are typically composed of monocrystalline, have a maximized shunt resistance, and a minimized series resistance. There are a few variations of PV cell chemistry, but the two most widely known are monocrystalline and polycrystalline silicon, whose names refer to the structure in which the silicon atoms are arranged. Monocrystalline has a higher average efficiency ranging from 15-20%, while polycrystalline has an efficiency of about 12-16%.

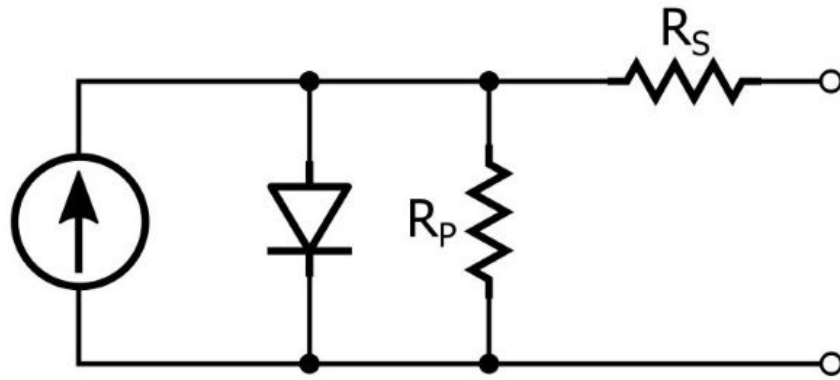


Figure 3.1: Equivalent Circuit of Solar Panel

The operating voltage and efficiency of a solar cell can be affected by numerous things including temperature, irradiance, load conditions and the chemical makeup of the cell. A Maximum Power Point Tracking (MPPT) algorithm addresses this by varying the panel operating voltage to shift the operating point on the P-V curve towards its maximum power point. The panel voltage is adjusted by varying the duty cycle of the converter so that the load impedance seen at the panels' terminals is at $R_T = (V_{MPP}^2 / P_{MPP})$, the equivalent resistance of the solar panel. The terminal resistance is found to be $R_T = R_L / D^2$ for the buck mode and $R_T = R_L / (1-D)^2$ for the boost mode, where R_L is the actual load resistance and D is the current duty cycle ratio. Efficiency may go up to 25% by implementing a Maximum Power Point Tracking algorithm [1].

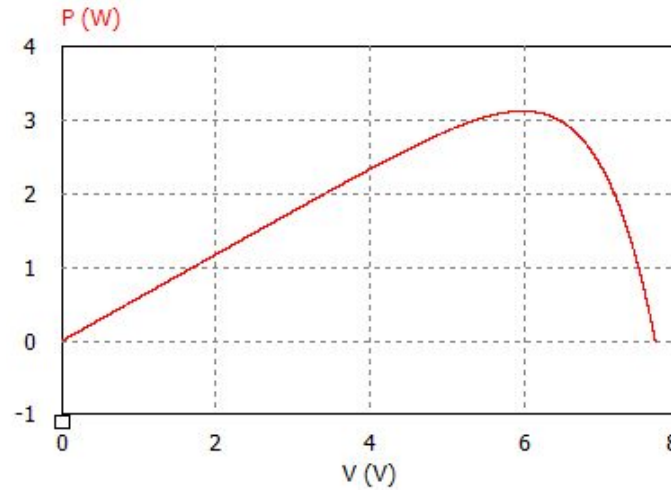


Figure 3.2: P-V curve of 6V, 3.4 W solar panel

Because the duty cycle of the converter directly affects the output voltage and current, the algorithm would take second priority in the loop regulation and will only be utilized when the panel cannot provide sufficient energy to properly charge the battery or power the mote. The adjustments to the duty cycle should be small enough to not cause a drastic change on the output yet frequent enough to drive the panel to its desired operation power point. The actual P-V and I-V curve of the solar panel used for this project is shown in Figure 3.5 and Figure 3.4. The 6 V, 3.4 W monocrystalline solar panel used in this project's design is also shown in Figure 3.3. Compared to the theoretical P-V curve estimated in Figure 3.2, the actual P-V curve has a lower maximum power. This was because the conditions in which the data was recorded caused partial shading or had insufficient brightness. While more efficient, monocrystalline cells also have a more drastic response to partial shading which must be considered when implementing the MPPT algorithm.



Figure 3.3: Measuring Panel Voltage at Varying Loads

I-V Curve

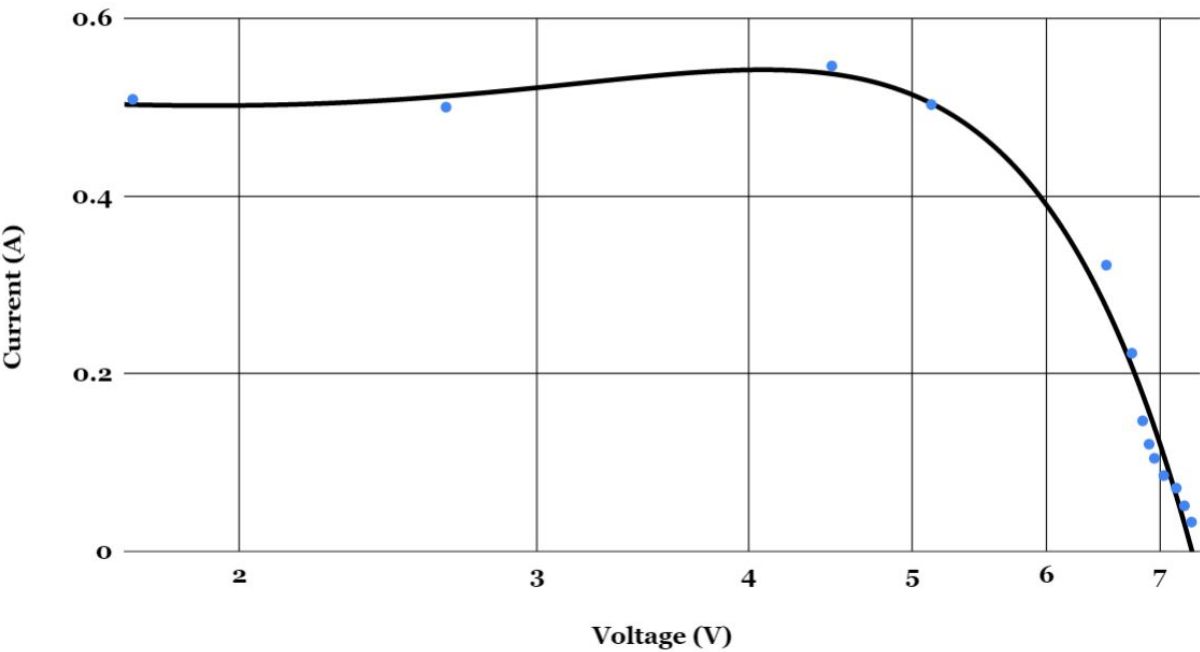


Figure 3.4: I-V Curve of 3.4 W, 6 V Adafruit Solar Panel

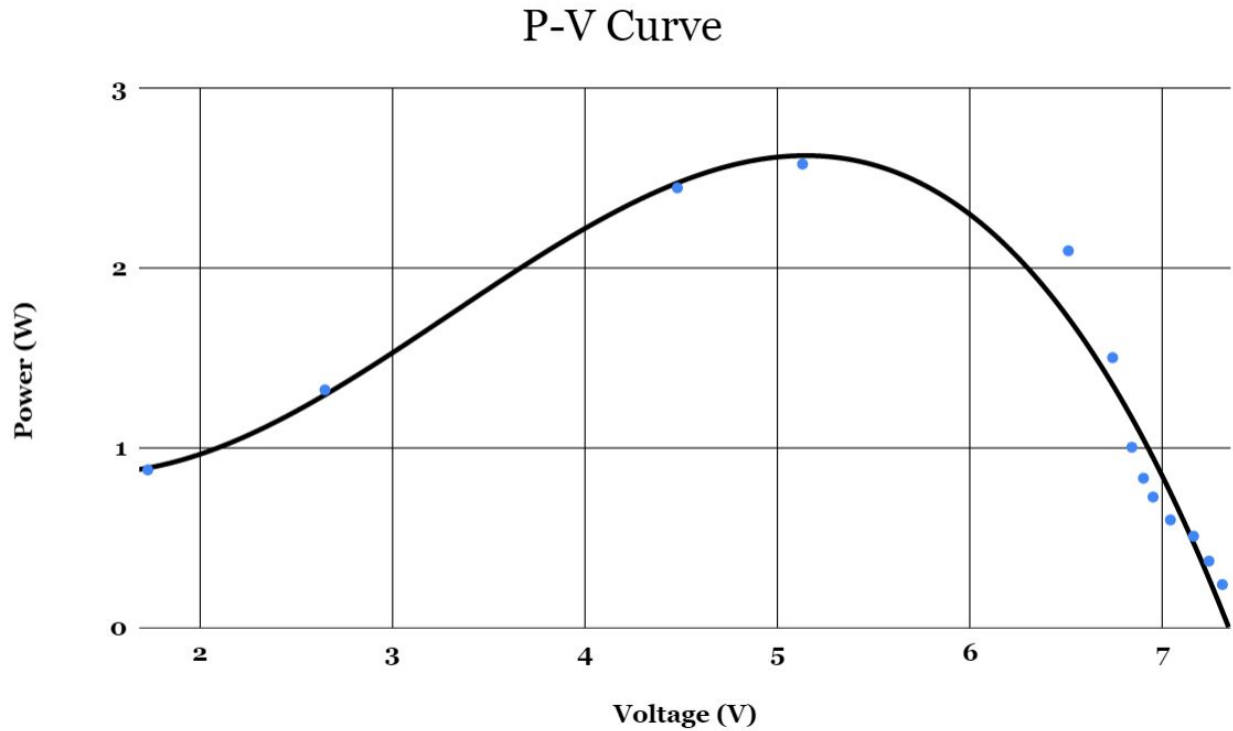


Figure 3.5: P-V Curve of 3.4 W, 6 V Adafruit Solar Panel

Fortunately, MPPT can improve the panel response to partial shading. Additionally, bypass diodes placed in parallel with the solar panel can improve partial shading effects by providing a current path around a shaded cell on the solar panel [16]. This ensures that the current flowing through a well-exposed cell does not go through a shaded cell which could otherwise damage it.

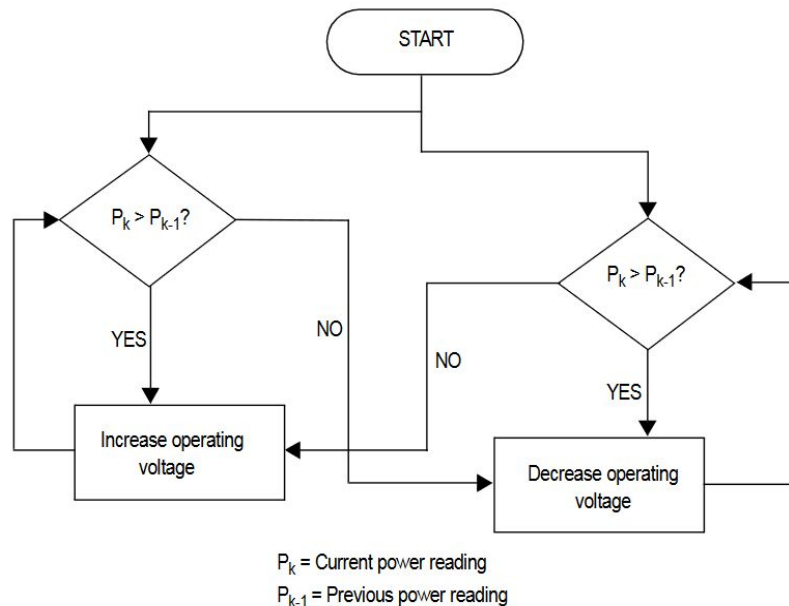


Figure 3.6: Perturb and Observe Algorithm Flowchart

There have been multiple MPPT algorithms; however, the most common are Perturb and Observe and Incremental Conductance. Perturb and Observe has the simplest implementation. If the current power is greater than the previous one, the algorithm continues to perturb the solar panel voltage in that direction by adjusting the duty cycle. For example, using the P-V Curve in Figure 3.5, if the solar panel voltage is currently at the right side of the curve, the algorithm will eventually change the duty cycle so that the operating voltage of the panel is shifted to the left towards the highest point on the curve. The larger the steps to the MPP, the faster it will achieve the solar panel's maximum power; however, large steps may also cause a constant oscillation around the MPPT [3]. The Perturb and Observe algorithm was chosen for this project due to the low processing speeds it would require for a low power system.

3.3 Power Management Design

3.3.1 Power Stage

A two switch buck-boost converter (figure 3.6) was chosen for its ability to handle a wide range of input voltages while maintaining a constant output as well as its relatively simple design that requires fewer components than the Cuk or Zeta converter topologies [12]. Buck-boost operation is necessary as the voltage from the panel is non-linear in relation to the power drawn from it, and can vary drastically depending on environmental and load conditions. The converter will switch between buck mode (duty applied to Q1, Q2 is open) or boost mode (Q1 is always closed, duty is applied to Q2) but never buck-boost mode where both switches are being actively driven. This minimizes switching losses in the circuit and the stress on the MOSFETs, while also greatly simplifying the converter's controller design [1]. The general transfer function for the circuit

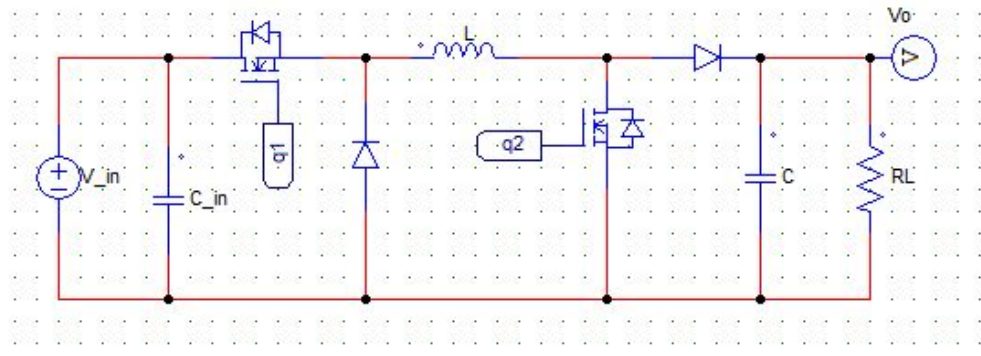


Figure 3.6: Two-Switch Buck-Boost Converter

The power stage was designed to keep a tight regulation on the voltage and current ripple in order to maximize battery safety and efficiency. To minimize the size of components and reduce the voltage and current ripple on the output, a switching frequency of 100 kHz was chosen. The maximum duty cycle D_{Max} was derived from analyzing the converter under its most strenuous

conditions; when the output voltage is at its max, with the minimum input voltage available in boost mode. The DC transfer function can thus be simplified to:

$$\text{Equation 3.1: } V_O = V_{IN} \cdot \frac{d_1}{1-d_2}$$

Where d_1 and d_2 are the duty cycle ratios applied to switches Q1 and Q2 respectively. From this, we determined the maximum duty cycle about in boost mode to be about 20% with an input voltage of 3.0V, and an output of 4.2V. The maximum output current $I_{O, \text{Max}}$ occurs at this duty value, which gives an average inductor current value of about 750 mA. From this, we can derive the formulas shown below, an inductor value of 100 μ H was chosen to keep the inductor current ripple under 10% at a maximum current output current of 600 mA. An output capacitor value of 700 μ F was derived to reduce the voltage ripple to < 1%, which is necessary when charging Li-ion batteries. The design opted for a 820 μ F capacitor. The design opted for a 820 μ F capacitor, since it offered a lower equivalent series resistance (ESR), minimizing power loss.

$$\text{Equation 3.2: } L = \frac{(V_{IN} - 2 \cdot V_{SW}) D_{Max}}{f_s \cdot \Delta I_L} = 80 \mu H, \quad V_{SW} \approx .05 V, \quad f_s = 100 \text{ kHz}$$

$$\text{Equation 3.3: } C = \frac{I_{O, \text{Max}} \cdot D_{Max}}{f_s \cdot (\Delta V_O - \Delta I_{L, \text{Max}} \cdot R_{ESR})} \approx 700 \mu F, \quad R_{ESR} \approx 50 m\Omega$$

Low voltage 19TQ015 Schottky diodes were also chosen for their low forward voltage drop in order to maximize energy transfer. Schottky diodes with a less than 200mV forward voltage were chosen. High-switching speed IRFZ14N mosfets were chosen for their relatively low turn-on voltage ($\sim 1V$) and their low drain-to-source resistance ($r_{DS} \approx 50m\Omega$).

The converter will be controlled in a closed-loop configuration using the PIC16F1778 microcontroller and an type-III analog compensator. The microcontroller will also implement the MPPT algorithm as well as the CC/CV charging algorithms.

3.3.2 Energy Storage

The system was planned to store its energy using a 1100 mAh polymer Li-Ion battery, which would provide power for the sensor mote when there wasn't adequate power being provided from the solar panel. A polymer Li-Ion battery was chosen for its high energy density and relatively low cost. In order to maximize the batteries capacity and longevity, it requires a two stage (sometimes 3) constant current/constant voltage charging process demonstrated in Figure 3.6. In this method, the battery is charged at a constant current, typically 0.5C A (half the maximum capacity) until it reaches its maximum voltage, typically 4.2 V. Once it reaches the maximum voltage, the charger should switch into constant voltage mode. Both phases require the

applied current or voltage to be within 1% error. The battery has completed charging once the battery current goes down to $0.1C$ and the battery voltage is at maximum. Both current and voltage control can be accomplished by adjusting the reference voltages of the converter's voltage feedback controller accordingly using samples taken from voltage or current sensors. When there is sufficient power being provided from the solar panel, the system will redirect that power to the sensor module to perform its operations. This way, the system will minimize its use of the battery, extending the batteries life.

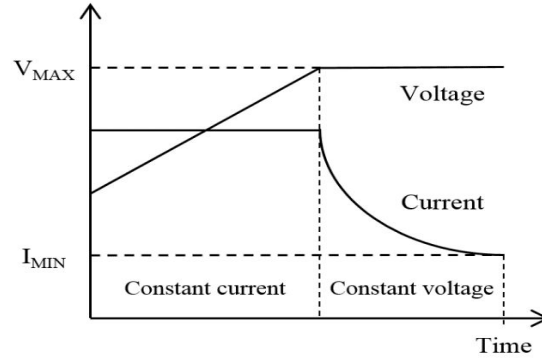


Figure 3.6 CC/CV charging scheme

researchgate.net/figure/Constant-current-constant-voltage-charging-profile-for-a-single-battery-cell_fig6_290358853

It is also necessary to monitor the amount of charge that has been applied to the battery and its estimated state of charge (SOC) in order to avoid damage to the lithium ion battery. It is generally best not to fully charge a lithium battery to avoid the risk of damaging it. The simplest way of determining the state of charge is through the use of coulomb counting. Coulomb counting integrates the battery current in order to determine the state of charge of the battery as seen in equation 3.4 where I_b is the battery current, I_{loss} is the current consumed by loss reactions, C_{rated} is the capacity of the battery in Ah, and $SOC(t_0)$ is the initial state of charge.

Equation 3.4:

$$SOC = SOC(t_0) + \frac{1}{C_{rated}} \int_{t_0}^{t_0 + \tau} (I_b - I_{loss}) dt$$

However, the initial state of charge must be determined beforehand in order to have an accurate starting point for the charging algorithm. If enough time has elapsed (approximately 4 hours), the SOC of the battery could be determined by using the open circuit voltage of the battery since it is proportional to its state of charge. If all the battery properties are provided, PSIM is capable of providing the proportional relationship between the two as seen in Figure 3.7. It is also necessary to note that as the constant current used to charge the battery decreases, the longer the battery will take to reach the maximum voltage, saturation, and therefore constant voltage mode.

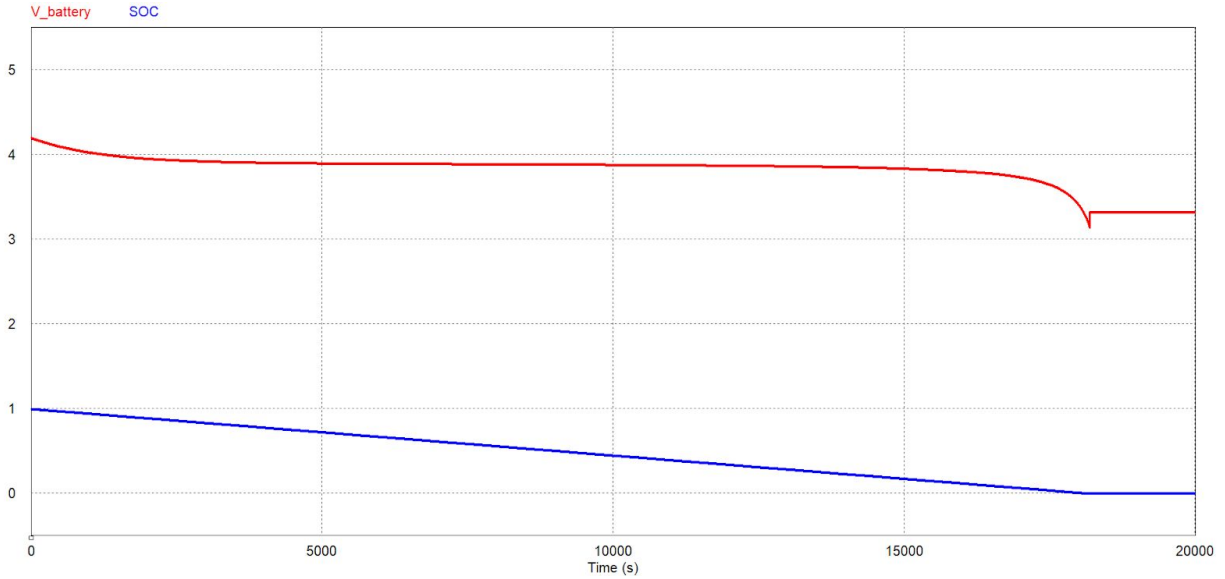


Figure 3.7: The SOC of the battery & terminal voltage v.s. time in seconds (discharging)

Figure 3.8 shows the proposed flowchart of the intended charging scheme of the battery within the microcontroller. The microcontroller could implement the closed loop control for the DC/DC converter using its internal op-amps to implement an error amplifier and a type-III compensation controller. The controller is further discussed in section 3.3.4.

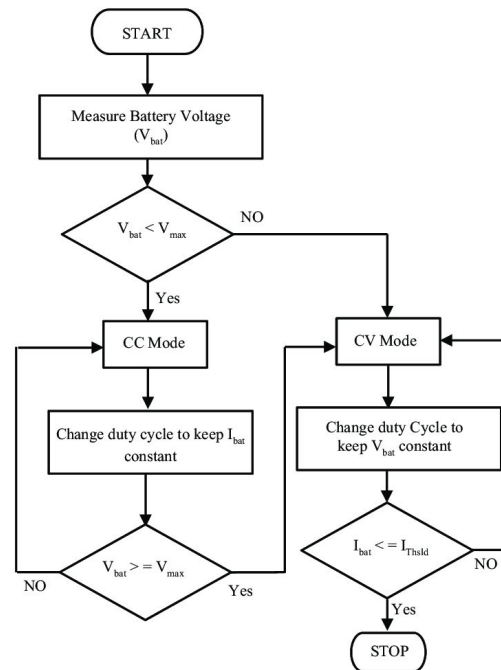


Figure 3.8 The flow chart of the charging control algorithm.

3.3.3 Control Design

Shown in Figure 3.9 is the proposed controller design for the two-switch buck-boost converter. The simplified C block (SSCB) was used to implement the functions of the microcontroller for the PSIM simulations. The output voltage of the converter is compared to a reference generated by the control block 'Regulator'. The error voltage then passes through the type-III compensator and through the limiter. The control signal from the limiter then goes into a comparator which generates the PWM control signal using a 100 kHz 1.8V amplitude saw-tooth carrier wave. The converter mode is determined by comparing the output and input voltages, which determines which switch is actively controlled.

$$\text{Equation 3.5: } G_c(s) = \frac{K_c}{s} \frac{(1+sTz1)^2}{(1+sTp1)^2} = \frac{1540}{s} \frac{(1+s/780)^2}{(1+s/1.4e5)^2}$$

The design will utilize a single voltage control loop in order to minimize the complexity of the circuit. It is necessary to design the controller so that the system is stable at its the extreme operating conditions. The power stage transfer function $\frac{v_o(s)}{d(s)}$ is shown in Equation 3.6. Figure 3.10 shows the power-stage bode plot under varying load and input voltage conditions. The phase margin for the controller was set in the range of 45°-50° with a cutoff frequency of 1.8kHz. The power stage transfer function is given in Equation 3.2, which considers the DC resistance of the inductor R_L and the ESR of the capacitor r_C . R_O is the output load resistance of the converter, $V_{O,max}$ is the max DC output voltage (4.2V), I_{in} is the corresponding DC input current and D is the corresponding duty cycle.

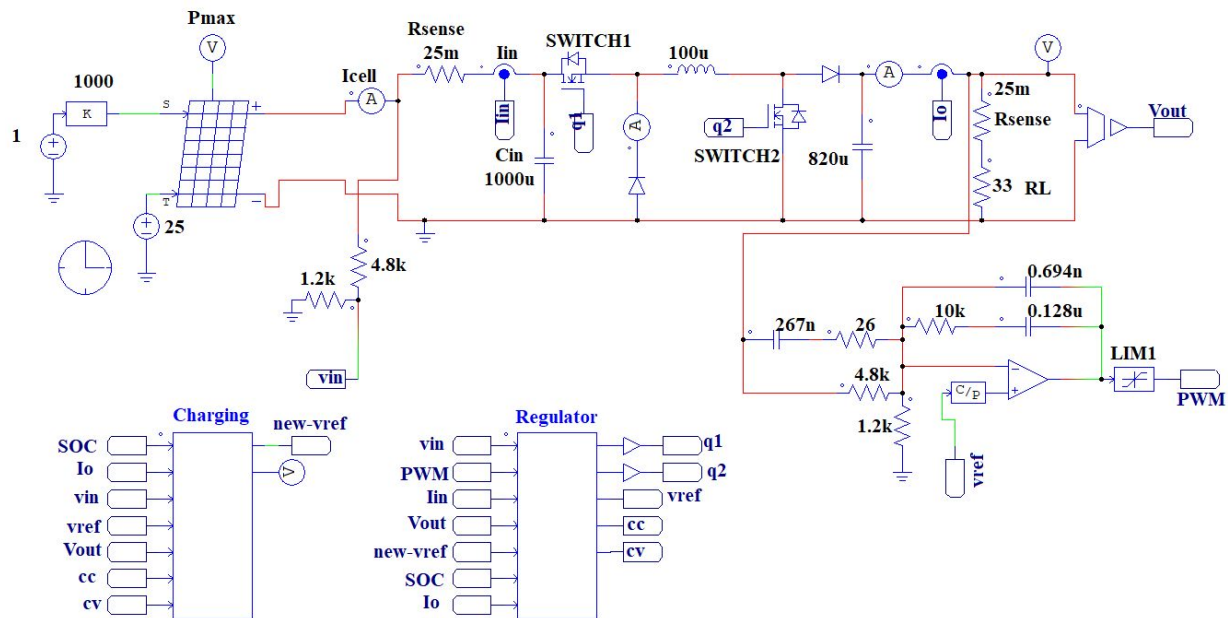


Figure 3.9: Proposed system design, designed in PSIM

Equation 3.6: $G_{PS}(s) = \frac{v_o(s)}{d(s)} = \frac{V_{O,max} - (Z_L \cdot I_{in}) / (1-D)}{(1-D) + Z_L / [Z_O \cdot (1-D)]}$, $Z_L = sL + R_L$, $Z_O = \frac{R_O(1+sCr_C)}{1+s(R_L + r_C)C}$

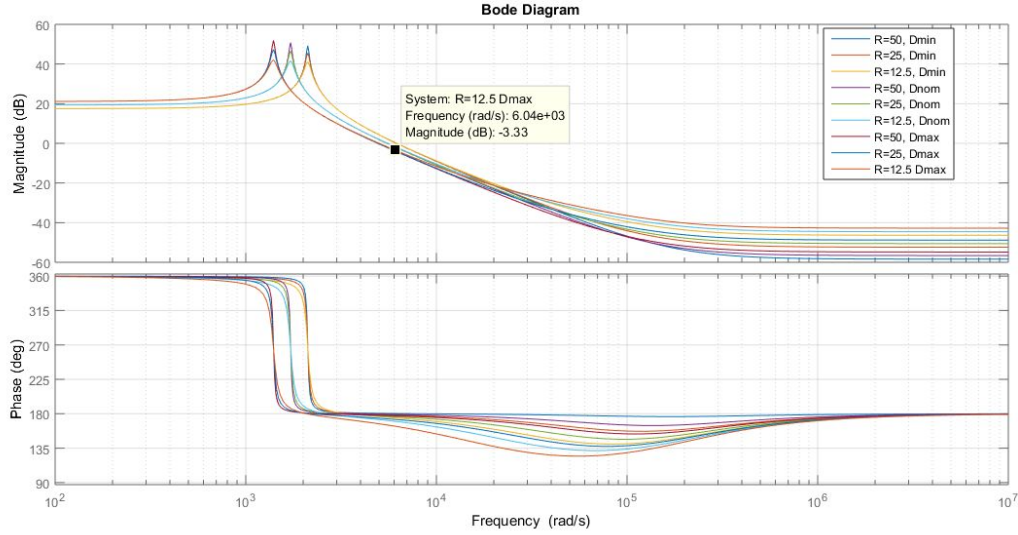


Figure 3.10: Power stage transfer function based on varying operating conditions.

The power stage requires a rather large phase boost past 1000 rad/s so a type-III compensator is chosen in order to provide a 150° phase boost. The compensator is implemented using an RC network and an op-amp acting as a differential amplifier (Figure 3.11).

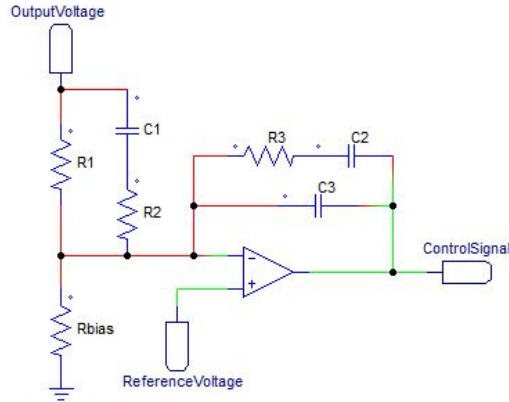


Figure 3.11: The type-III compensator implemented using an op-amp.

The compensator provides the control signal which is used to derive the duty ratio for the circuit. The resistor and capacitor values were derived using the K-factor approach detailed in *The K Factor: A New Mathematical Tool for Stability Analysis and Synthesis* [2]. The value of R_{Bias} (chosen to be 1.2kΩ) has no effect on the small signal transfer function of the controller, but

introduces the scaling factor K_{FB} which appropriately scales the output voltage by 0.2. The controller was then tuned using the Control System Tuner in Matlab, yielding the transfer function given in Equation 3.7.

$$\text{Equation 3.7: } Gc(s) = \frac{K_c}{s} \frac{(1+sTz1)^2}{(1+sTp1)^2} = \frac{1540}{s} \frac{(1+s/780)^2}{(1+s/1.4e5)^2}$$

It was then converted to the Z-domain using Matlab's `c2d` function. The bode plot of the controller is plotted in Figure 3.12 along with its digital, Z-domain FOH equivalent. The Z-domain equivalent appears to behave like its analog counterpart up to 10^5 rad/s. If the design proves to be successful, we hope to implement the compensator strictly using the Z-domain equivalent using the microcontroller to minimize the component count. Figure 3.13 shows the power stage transfer and controller's transfer function, plus the compensated system adjusted for the appropriate phase margin.

$$\text{Equation 3.8: } Gc[z] = \frac{49.13 z^{-3} - 16.17 z^{-2} - 116.2 z^{-1} + 83.27}{1 + (-1.916 z^{-1} + 1.126 z^{-2} - 0.2098 z^{-3})}$$

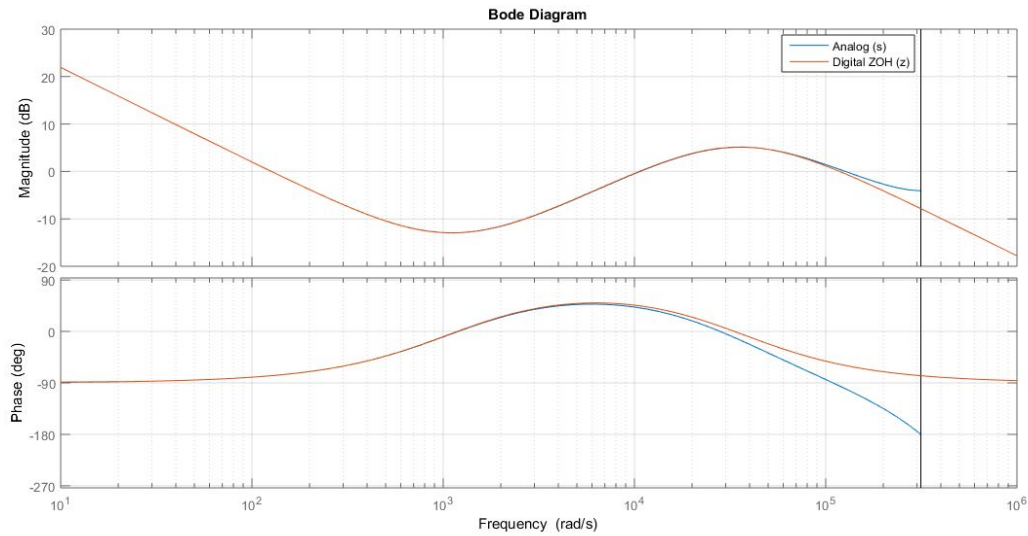


Figure 3.12: The transfer function of the controller and its Z-domain equivalent

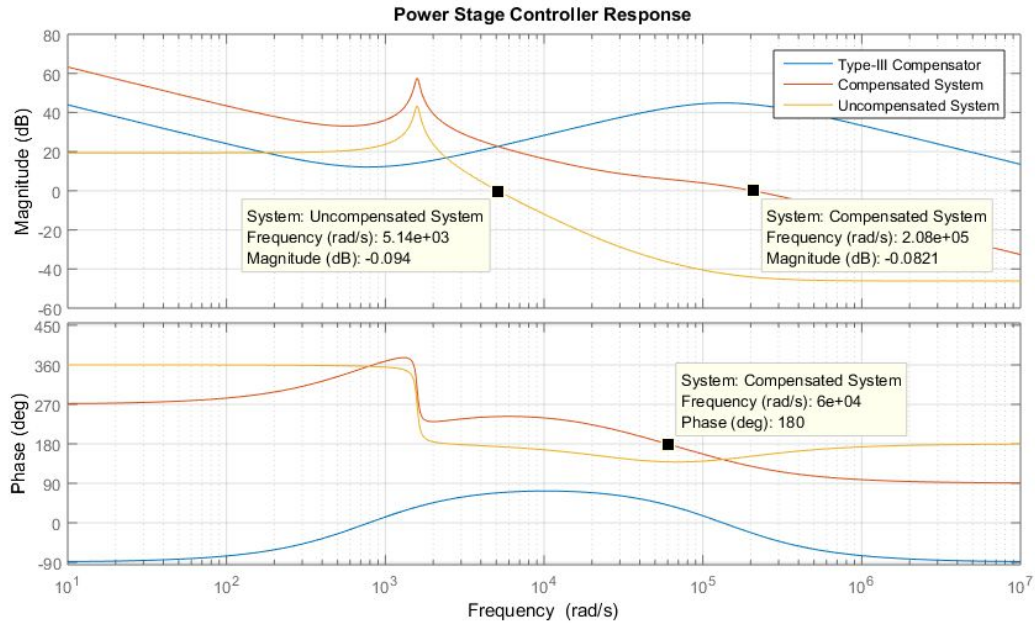


Figure 3.13: The power stage uncompensated and compensated

3.3.4 Hardware Implementation

The circuit layout is shown in figure 3.16. The circuit's voltage and current signals were buffered using a quad-package operational amplifier (LT6005) to improve signal integrity. When tested in buck mode, $V_{IN} = 5.06V$ with a duty ratio of 0.50, it was found that the MOSFET switching action introduced large voltage spikes on the output, in addition to the expected voltage ripple. These spikes could be reduced by adding a decoupling capacitor to the gate of each MOSFET though this will introduce an undesirable phase delay at in the PWM signal. Instead an RC snubber circuit is added in parallel to the output MOSFET Q2. This reduced the voltage spikes from a value of $1.41V_{P-P}$ to $760mV_{P-P}$. The snubber circuit does introduce an increase in the inefficiency of the system but only by a negligible amount. The resistor and capacitor values were derived using the formulas 3.9 and 3.10. $V_{O,Max}$ is the peak value of the voltage spike (4.7V), t_f is the fall time of the MOSFET (40 nS, verified on the oscilloscope), V_{br} is the breakdown voltage of the diodes (15V) and f_s is the switching frequency (100 kHz). The values $C_S = 3.3nF$ and $R_S = 6\Omega$ were chosen.

$$\text{Equation 3.9: } C_S = \frac{I_{L,Max} \cdot t_f}{2 \cdot V_{O,Max}}$$

$$\text{Equation 3.10: } R_S < \frac{D_{Min}}{3 \cdot C_S \cdot f_s}, \quad R_S < \frac{V_{br}}{I_{L,Max}}$$

To measure the input and output currents, the MAX4172 IC was used. It utilizes the $25\text{m}\Omega$ sense resistor and an internal current mirror circuit to produce a voltage proportional to the current through R_{Sense} . The voltage is then sampled by the ADC module on the PIC16. The relationship between the current through R_{Sense} (I_{Sense}) and the IC's output voltage V_O is given by equation 3.11, where G_m is the transconductance of the op-amp internal to the IC.

$$\text{Equation 3.11} \quad \frac{V_O}{I_{\text{Sense}}} = G_m \cdot R_{\text{Sense}} \cdot R_O$$

This allows for the designer to set whatever gain they find appropriate by varying the values of R_{Sense} and the IC's output resistor R_{Out} , and account for this gain in the microcontroller's software. An output resistor value of $R_O = 12.84\text{k}\Omega$ was chosen for a target current gain of 0.312 so as to not exceed the voltage range on the PIC16's ADC module when the converter is operating with its maximum current. The IC was tested and the results, shown in figure 3.15, show the gain to be closer to 3.2. An issue did arise in which the accuracy of the sensor suffered when the current through R_{Sense} fell below 100mA . This can be fixed by increasing the value of the sense resistor or the value of the output resistor.

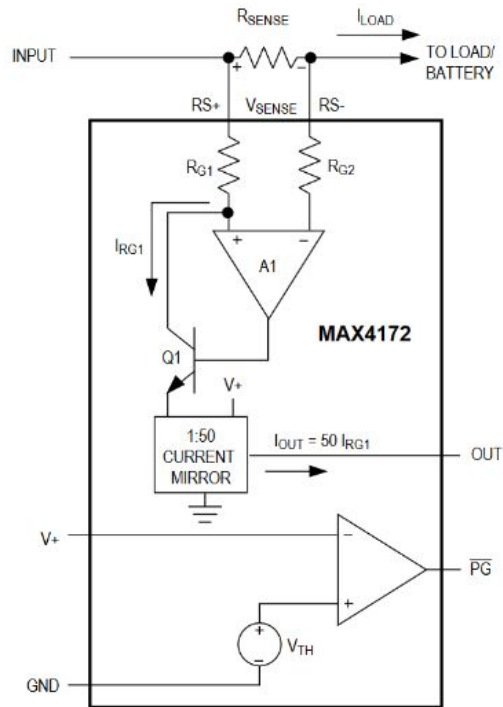


Figure 3.14: The internal schematic of the MAX4172 current sensing IC.

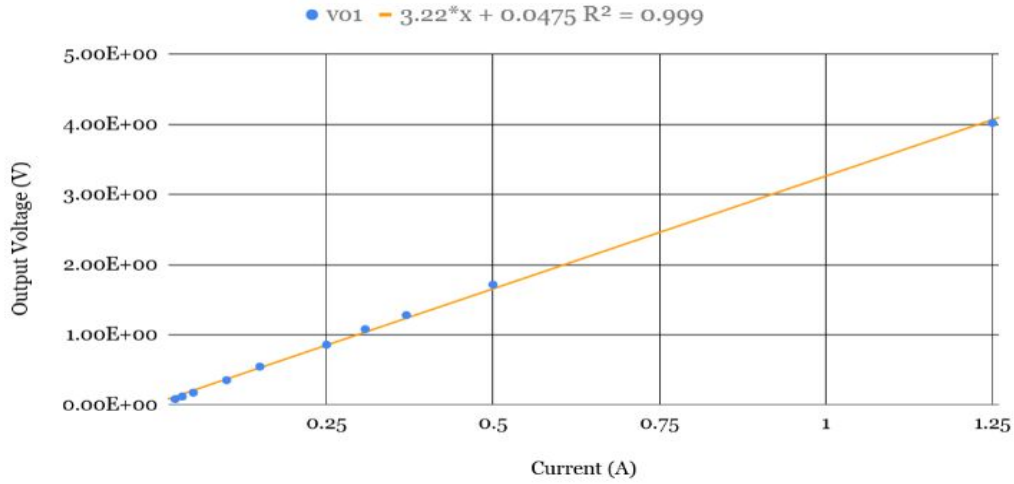


Figure 3.15: Voltage from the MAX4172 versus the current through R_{Sense} .

The MOSFET switches had been tested with a gate resistor greater than 1 k Ω which resulted in a heavily distorted PWM signal. A gate resistor of 100 Ω was chosen which provided reasonable stability with little effect on the MOSFETs response time.

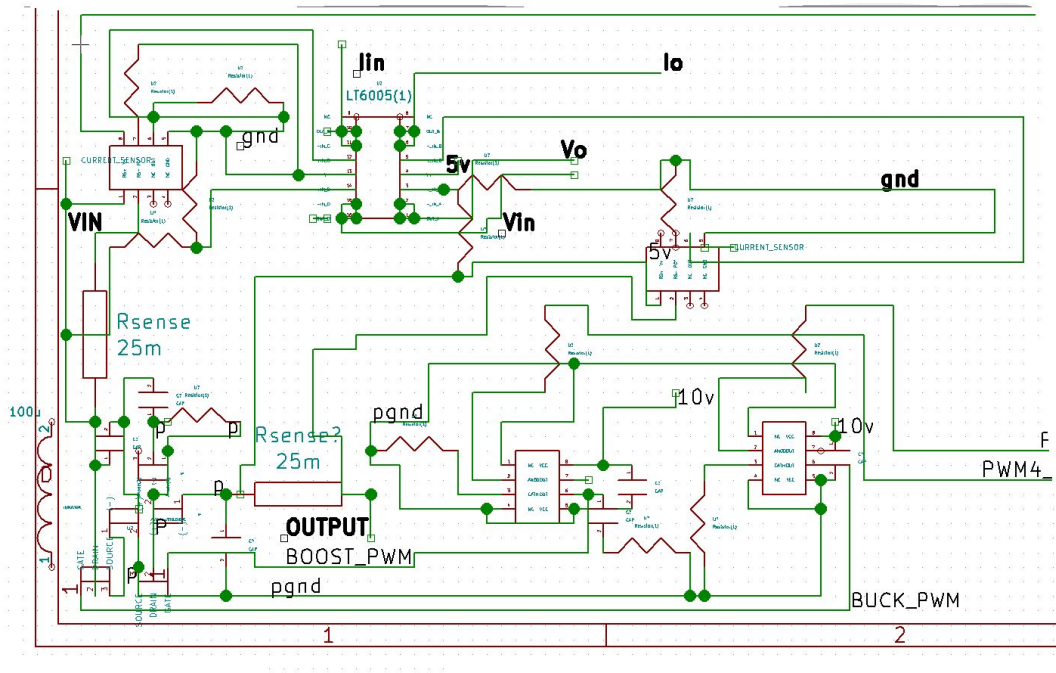


Figure 3.16: Physical layout of the proposed converter, designed in KiCAD

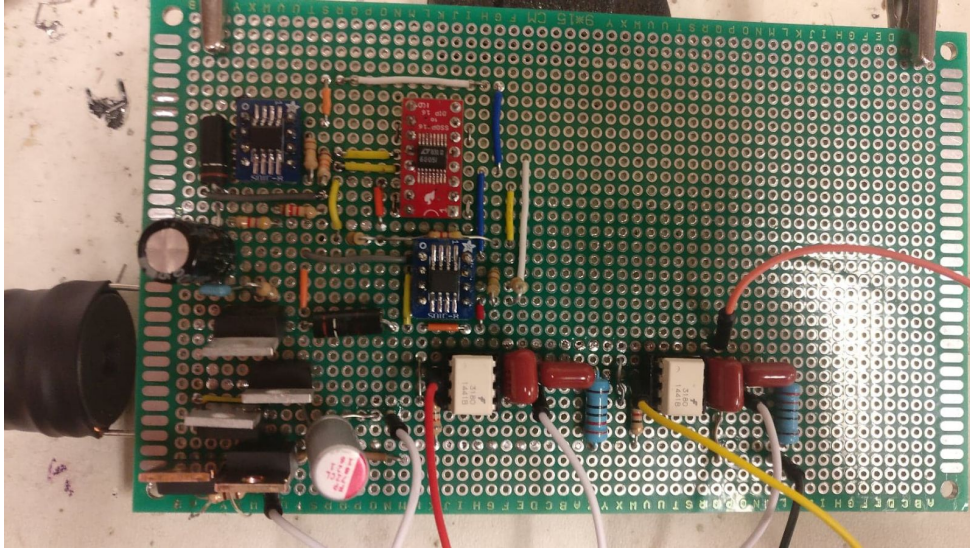


Figure 3.17: The final circuit design. The PIC16 is offboard and can be seen in figure 3.18

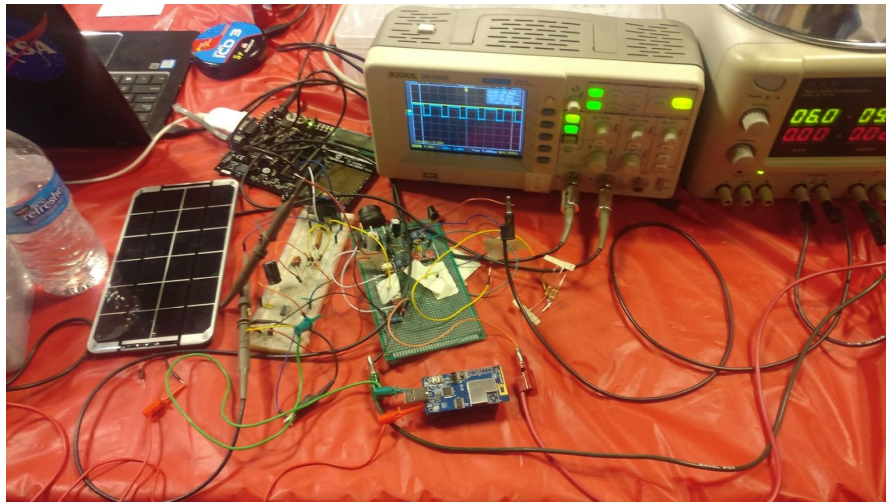


Figure 3.18: The system set up, including the sensor mote. The PWM signal and the output voltage of the converter are shown on the oscilloscope.

3.4 Simulation

3.4.1 Power Stage Simulation

The DC/DC converter was tested at an input voltage of approximately 5 V and a load resistance of 33 Ω . In order to achieve the reference voltage of 4.2 V, the converter went into buck mode. The inductor current, capacitor current, diode current, and inductor voltage are shown below. The minimum inductor current is about 0 A which means that the converter is approaching

DCM, but is hovering just above at the given load.

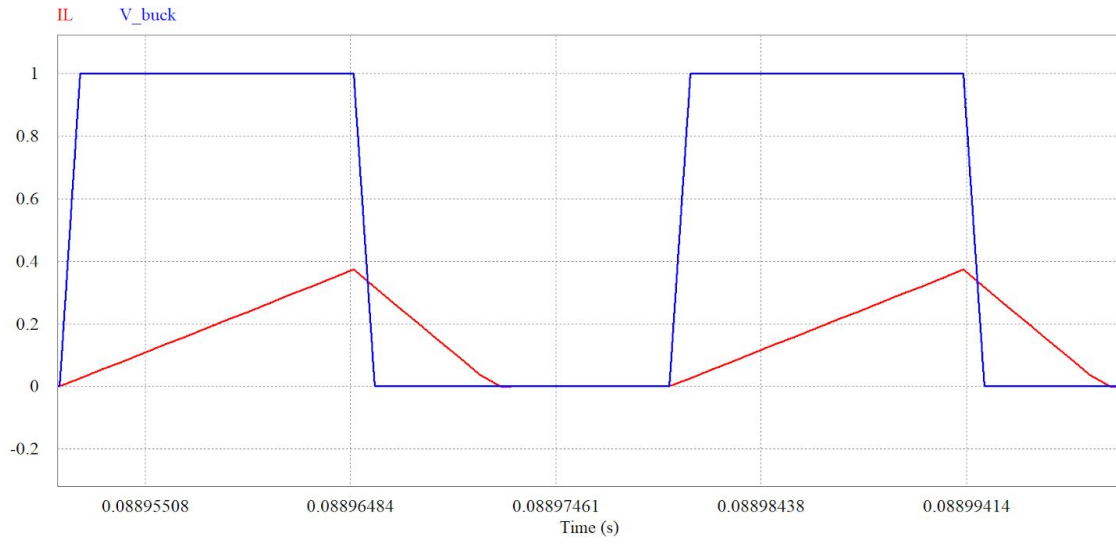


Figure 3.19: Inductor Current and Duty Cycle ($I_{Lmin} = 0$ A, $I_{Lmax} = 372$ mA)

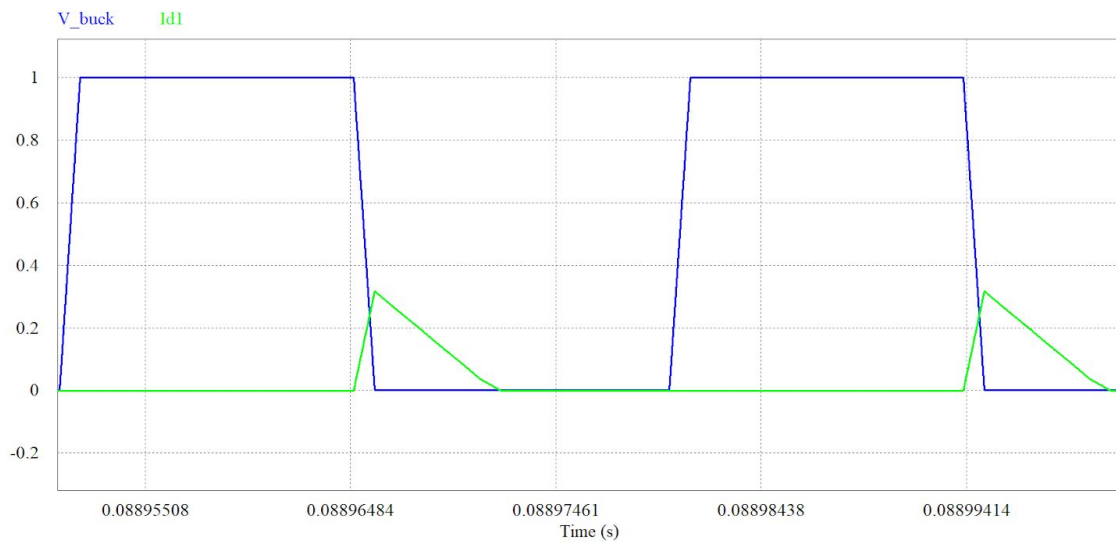


Figure 3.20: Diode Current and Duty Cycle ($I_{dmin} = -2.4\mu\text{A}$, $I_{dmax} = 310$ mA)

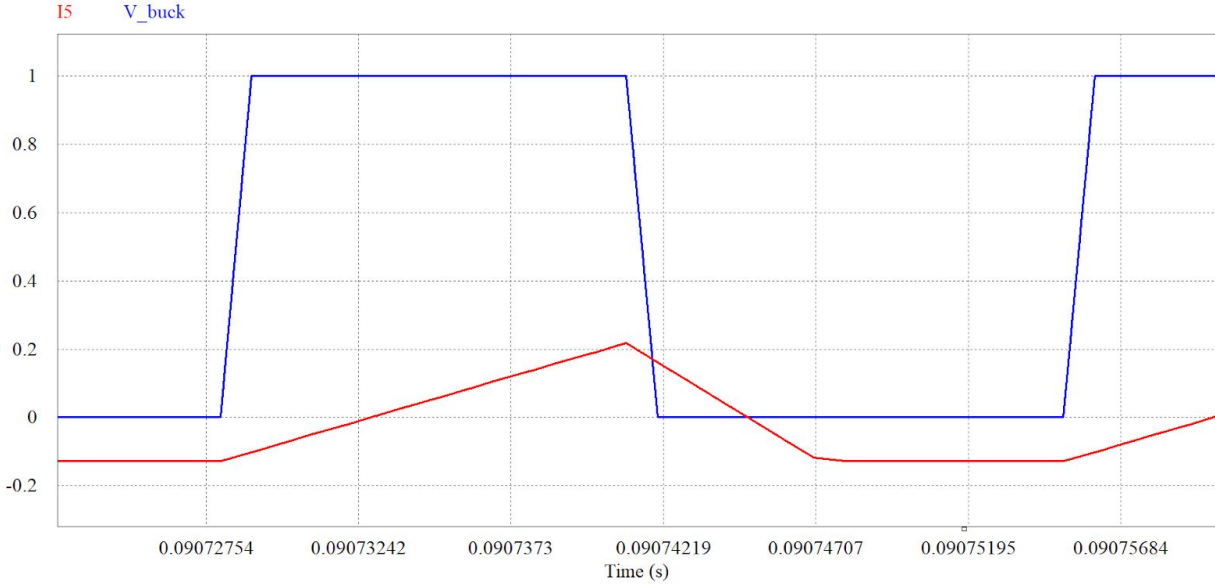


Figure 3.21: Capacitor Current and Duty Cycle ($I_{c_min} = -127 \text{ mA}$, $I_{c_max} = 215 \text{ mA}$)

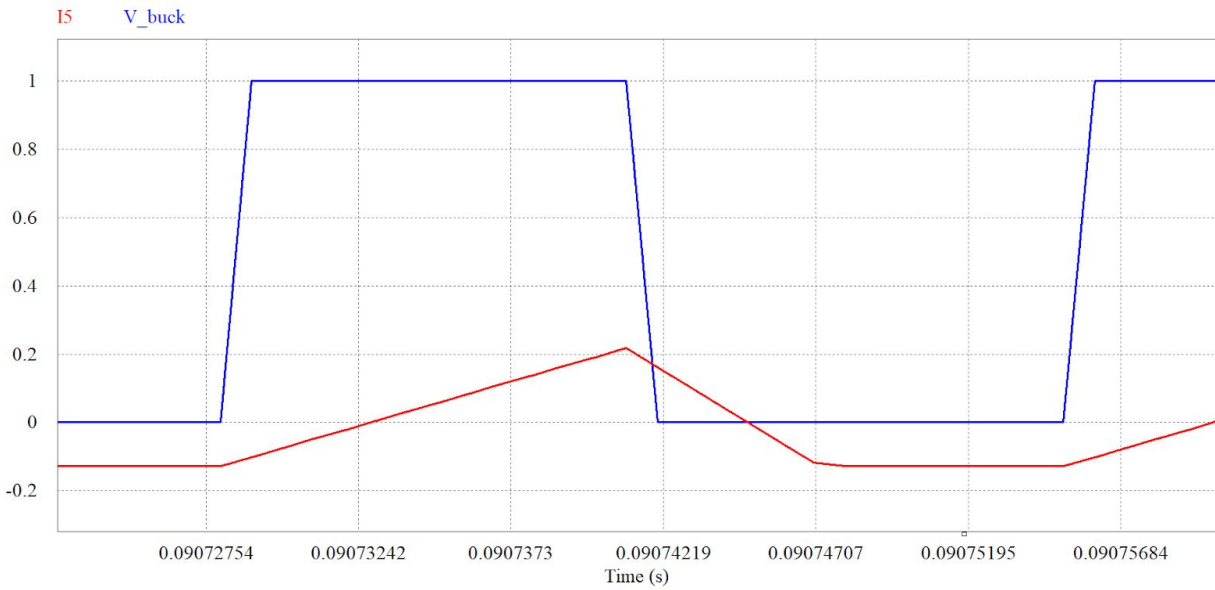


Figure 3.22: Inductor Voltage and Duty Cycle ($V_{L_min} = -0.128 \text{ V}$, $V_{L_max} = 0.243 \text{ V}$)

3.4.2 Compensation Results

After simulating a stable system response for the controller transfer function, the dynamic performance of the converter was tested in PSIM. The schematic used to model the controller and converter is shown in Figure 3.7. In figures 3.23a, 3.23b, 3.24a and 3.24b, simulation results for the output voltage and current are shown.

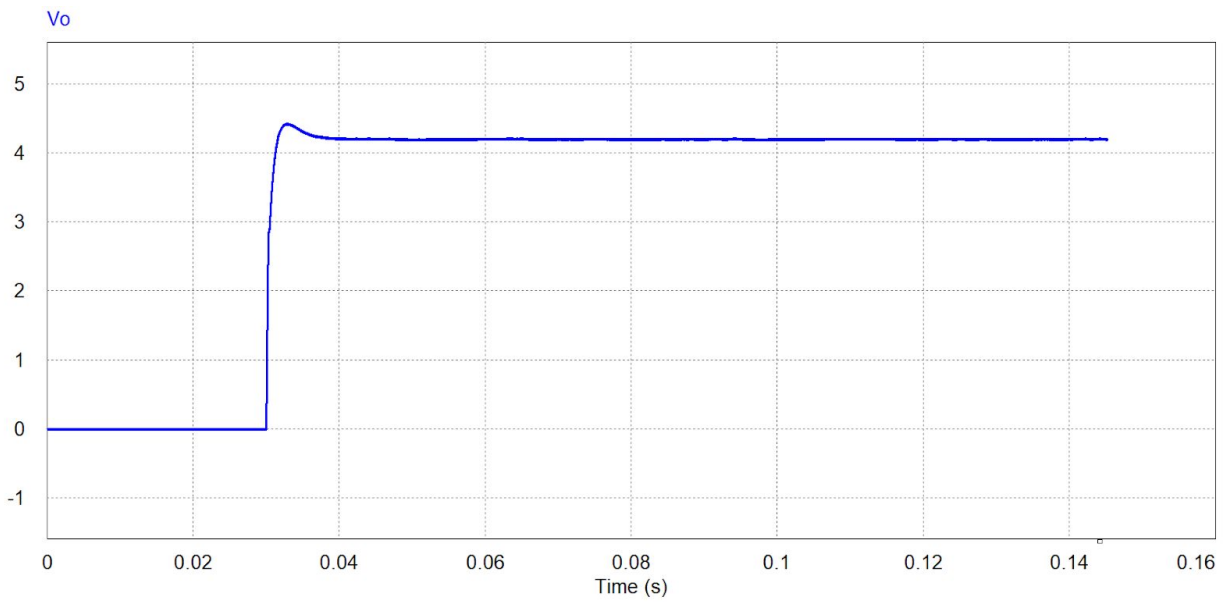


Figure 3.23a: Constant Voltage Simulation Results with compensator

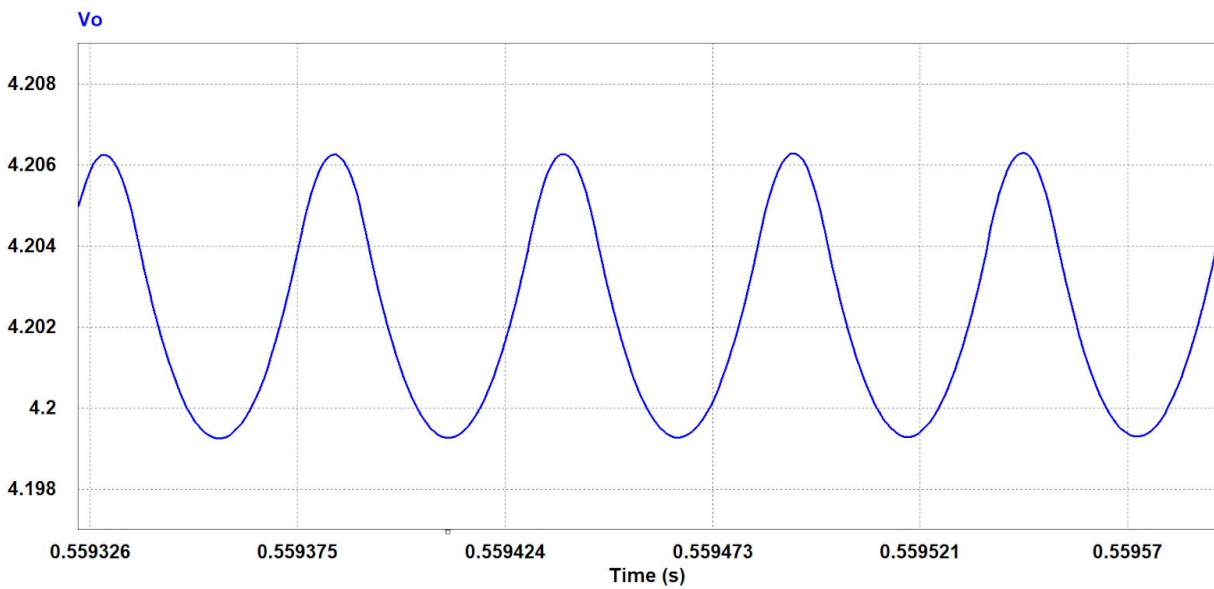


Figure 3.23b: Constant Voltage - output voltage ripple

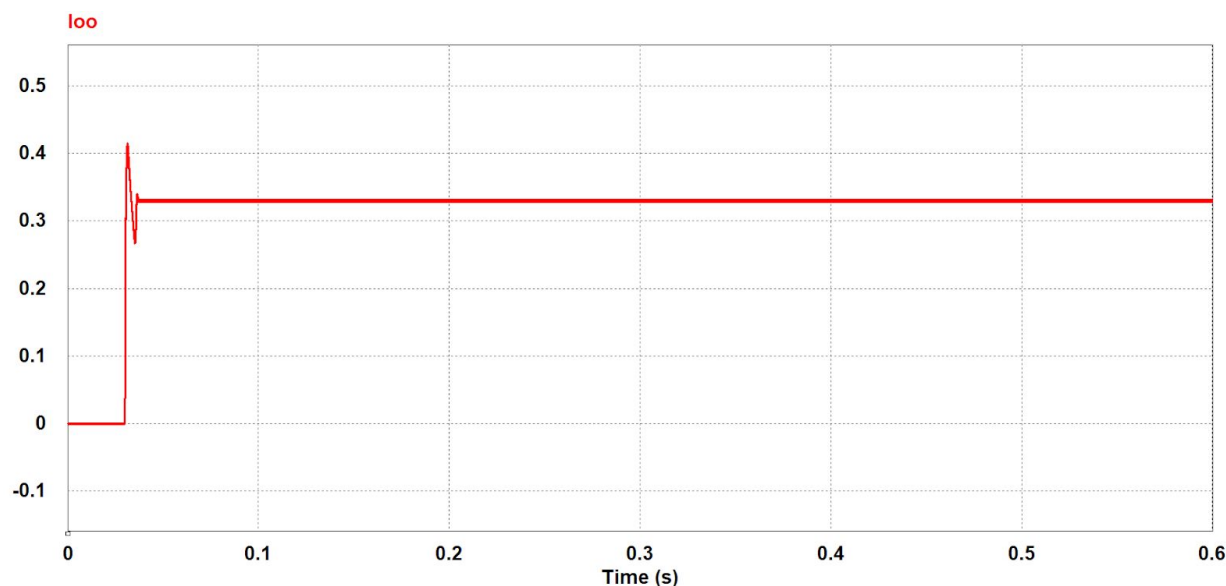


Figure 3.24a: Constant Voltage Simulation Results with compensator

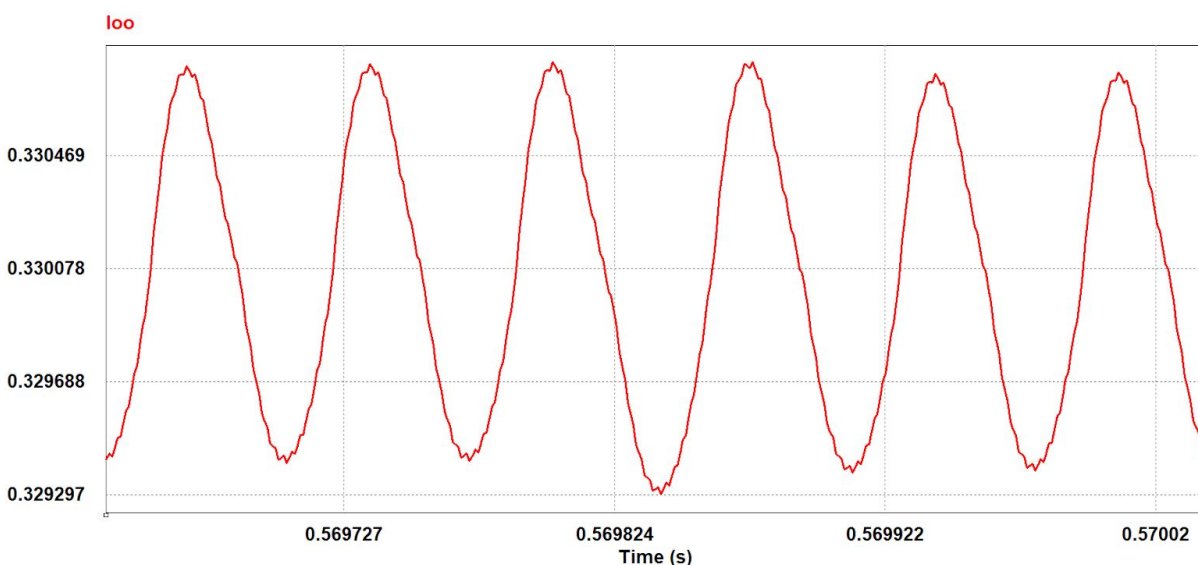


Figure 3.24b: Constant Voltage - output current ripple ($I_{avg} = 0.33$ A)

Below is also the result of MPPT tracking through the use of the Perturb and Observe Algorithm. Without taking into account the voltage or current limits of the battery, the maximum power point was successfully tracked as seen in Figure 4.25a and Figure 4.25b. MPPT is shown to use the input voltage (panel voltage) rather than the output voltage to achieve the solar panel's maximum power. As seen in Figure 2.25b, the algorithm oscillates around the maximum power as the irradiance of the simulated solar panel varies along with the panel's maximum achievable power.

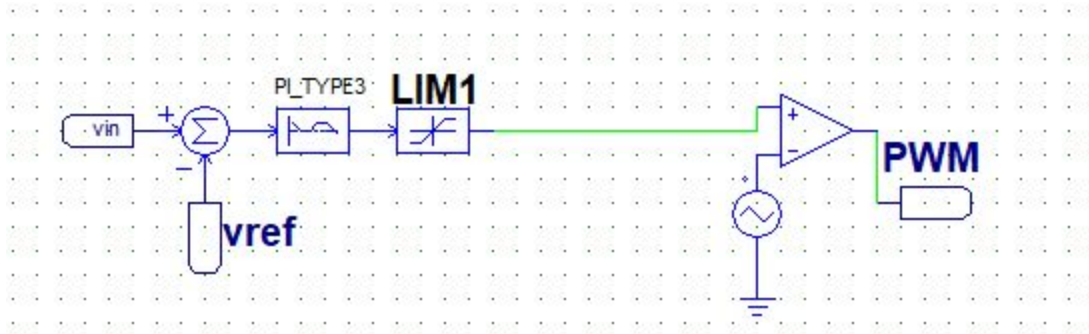


Figure 4.25a: Control Design for MPPT

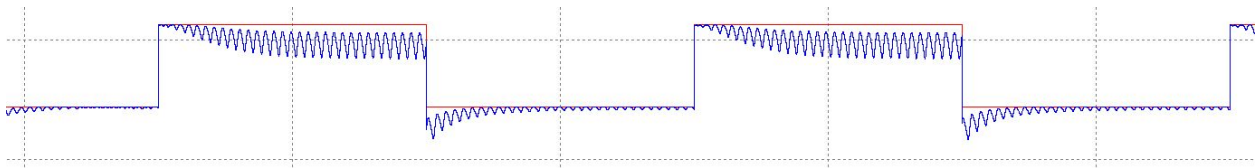


Figure 2.25b: Maximum Power Point Tracking with Perturb & Observe

3.5 Wireless Sensor Network

3.5.1 Wireless Sensor Node Choice

Of the different options for wireless sensor motes, Dr. Wang decided it was best to go with the TelosB sensor that we were leaning towards earlier in the project. Roughly twenty of these wireless motes were subsequently purchased. The TelosB wireless sensor mote is a great basic mote that has the most options for the cheapest price. It comes with a temperature and humidity sensor packed into one, two Hamamatsu light sensors (one for visible light, and another for infrared light), as well as a radio transceiver, two push buttons, three different colored LEDs, and finally a microcontroller to control it all. What is great is that as long as the software is understood, all of these can be manipulated to create a wireless sensor network once multiple nodes are talking to each other. Each of these motes costs about \$100, so it is not a cheap venture to say the least.

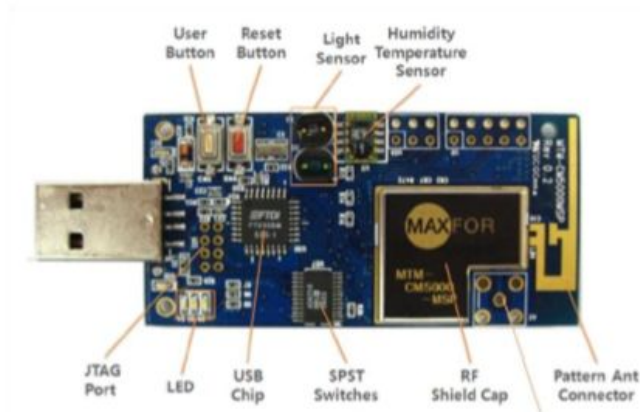


Figure 3.26: This is the top side of the TelosB mote.

3.5.2 TelosB Wireless Sensor Mote Component Breakdown

The temperature and humidity sensors are both packed into one device. They are from the SHT1x Sensirion family of sensors. The temperature aspect is measured using a band-gap sensor. This operates on the principle that temperature is dependent on the forward voltage of the silicon diode. The relative humidity is measured using a capacitive element. It is reliable from -40°F to 254.9°F .

The two light sensors are made by the Japanese company Hamamatsu Photonics. They are both encased in a ceramic housing to ensure that no light will enter from the sides or back of the sensor and alter the readings. The two sensors are the S1087(visible light sensor) and the S1087-01(infrared light sensor). They are able to make readings in low-light conditions as well as extremely bright conditions.

The RF Transceiver is a CC2420 chip made by Texas Instruments. It transmits wirelessly at 2.4 GHz and has an effective data rate of 250 Kb/s. This is more than sufficient for testing the mote to transmit data in a certain effective range. It should be able to transmit to a host and other motes in the area as well.

Finally the onboard microcontroller is a MSP430F1611 16-bit RISC microcontroller. It is specifically made for low-power applications and has five low power modes. It can even wake up from a low-power mode to an active mode in less than six microseconds. It also contains a fast 12-bit Analog to Digital converter as well as dual Digital to Analog converters.

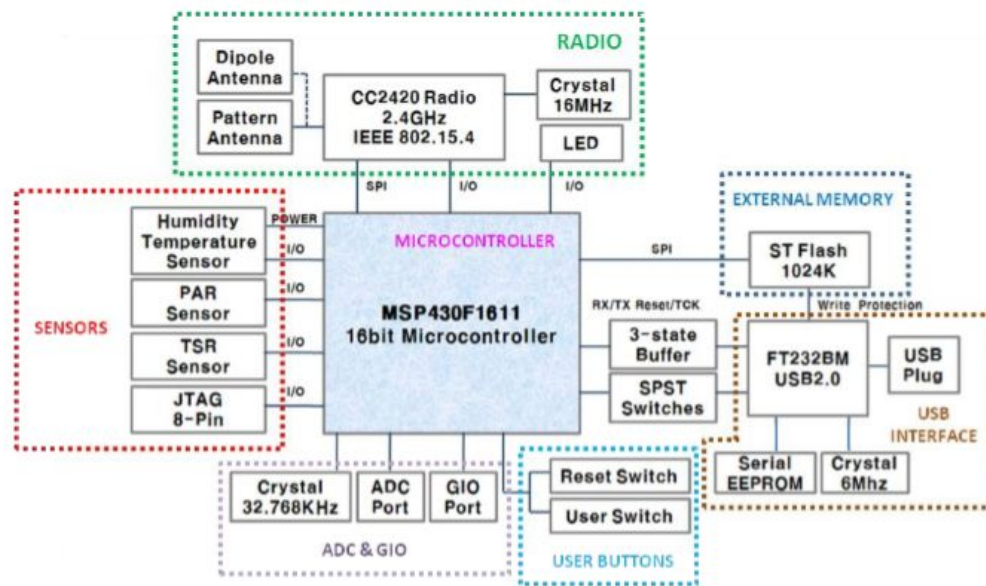


Figure 4: Sub-Component Diagram

Figure 3.27: This is the visual breakdown of the TelosB mote.

3.5.3 Contiki OS vs. TinyOS

In the context of actually utilizing the mote and being able to program it, a software had to be decided on to flash code onto the mote. During the initial research, TinyOS was discovered as the most popular option to go with. To be completely transparent it seemed like it was the only option that there was for quite some time. During the research, it was discovered that TinyOS is actually a headache to deal with. It had a few major problems plaguing it. First, it was extremely outdated. The files needed to run the software were either nonexistent or corrupted and that was from the original source. Secondly, there was almost zero documentation on how to use the TelosB mote with TinyOS once the software was downloaded. Finally, the download process itself was extremely convoluted and vague. There were explicit tool chains that were needed and a specific set of steps had to be taken so as to not download the files in the wrong order. This is when the idea of a virtual machine came into play.

VMWare is a program used to create and run virtual machines through virtual image files. This process is called virtualization. The TinyOS program was supposed to be run on VMWare using an image file downloaded from the source website. When it was accessed for the first time, the image file was corrupt and could not be read. That was when the option of using Contiki OS was discovered as a viable second solution. Contiki uses the same principle, but is a far better option. There were more than a few reasons to go with the program, but the main ones

are its quality, how updated it is and how reliable it can be. Once the image file was downloaded it immediately worked inside VMWare. The program was stable and allowed access to its terminal program and Cooja simulation program as well. Contiki OS in itself actually has a very clean interface and is updated on a regular basis. The overall reliability of this software trumped anything that TinyOS had going for it so that software was abandoned. At this point, however there was still an issue that needed to be addressed. There was hardly any documentation on how to actually use the TelosB mote with Contiki OS. After some research, a discovery was made that would quickly eliminate this problem.

3.5.4 Programming the Mote

So it turns out that the TelosB mote actually goes by a second name. It can also be called a Sky mote. This is due to the fact that two separate companies produced the same product, but wanted them to be different. So to solve this problem, Crossbow Technologies called their product a TelosB mote, and the other company called there's a Tmote Sky. The Tmote Sky was very popular, while the TelosB mote was somewhat neglected in the commercial space. However, the other company ended up dissolving so they stopped production of the Tmote Sky so now the only product left was the TelosB mote. The internet continued to buy the still available TelosB motes, but when writing documentation referred to them as a Tmote Sky because they were used to that name and the two motes are for the most part essentially the exact same. So once that was discovered a whole doorway of documentation was able to be accessed that was previously thought to be nonexistent.

To program the actual mote, you need to use the terminal program inside the Instant Contiki OS 2.7 software running on the virtual machine. Once there, you can access your .c scripts that have been written and attempt to change them into a .sky format for use with a Tmote Sky wireless sensor mote. The main process is to write a program, convert it to a .sky file, then flash it to the Tmote Sky. It was discovered that there were actually a folder accessible that held pre-made example code sections that could be used. Once that was discovered, the programming style was reverse engineered and a .c script was created using the specific structure that Contiki OS used and preferred. It was just a simple Hello-World program that flashed the LED's on the board in a certain order. The program should have been able to be programmed to the mote, but it would give errors every time it was attempted.

Then it was discovered that in order to run the code on a Tmote Sky, a set of files was needed to be downloaded that would add that capability to the users Contiki OS terminal program. The second problem was encountered when trying to actually upload the program to the mote. It claimed it did not have authority to access the space where the mote was plugged into the USB drive. So then the final piece of the puzzle was found when the sudo command was used to access the terminal in root. The command "sudo -s" needed to be entered every single

time the terminal was accessed to run it as root. The simple Hello-World program was then finally able to be programmed to the mote and the LED's flashed correctly.

The next big milestone was to test the actual sensors on the mote itself. Once another example program was found that used the sensors, the code could be reverse engineered into a small script that just gave readings from the temperature sensor, humidity sensor, and the light sensor. This whole program was then flashed over to the mote and executed as shown in Fig. 3.28. The program then gave readings just as expected and all three sensors were tested to the best of our ability. A light was cast onto the mote in Fig. 3.4.4.2 and that is the reason the lux measurement spikes from a reading of 35.41 lux to 221.05 lux. This was an indication among others that the sensors on that specific mote were working properly. This initial test will be conducted on every new mote we receive.



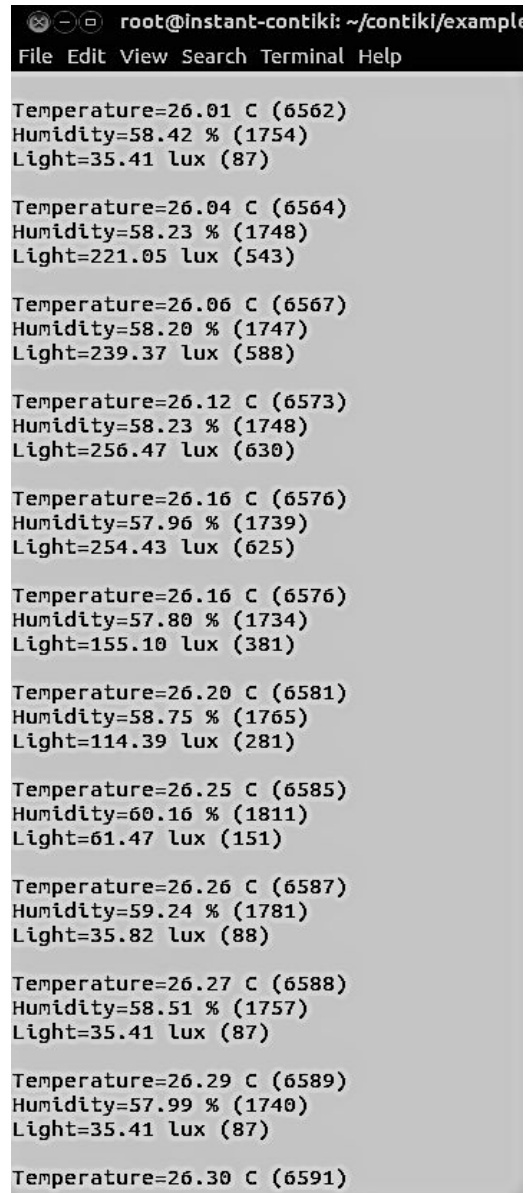
```

root@instant-contiki: ~/contiki/examples/rime
File Edit View Search Terminal Help
make[1]: Entering directory `/home/user/contiki/examples/rime'
make -k -j 20 sky-reset-sequence
make[2]: Entering directory `/home/user/contiki/examples/rime'
../../tools/sky/msp430-bsl-linux --telosb -c /dev/ttyUSB0 -r
MSP430 Bootstrap Loader Version: 1.39-telos-7
Use -h for help
Reset device ...
Done
make[2]: Leaving directory `/home/user/contiki/examples/rime'
make -j 20 sky-upload-sequence
make[2]: Entering directory `/home/user/contiki/examples/rime'
++++ Erasing /dev/ttyUSB0
MSP430 Bootstrap Loader Version: 1.39-telos-7
Use -h for help
Mass Erase...
Transmit default password ...
++++ Programming /dev/ttyUSB0
MSP430 Bootstrap Loader Version: 1.39-telos-7
Invoking BSL...
Transmit default password ...
Current bootstrap loader version: 1.61 (Device ID: f16c)
Changing baudrate to 38400 ...
Program ...
25642 bytes programmed.
++++ Resetting /dev/ttyUSB0
MSP430 Bootstrap Loader Version: 1.39-telos-7
Use -h for help
Reset device ...
Done
make[2]: Leaving directory `/home/user/contiki/examples/rime'
make[1]: Leaving directory `/home/user/contiki/examples/rime'
rm sensor-aq.ihex
root@instant-contiki:~/contiki/examples/rime# make login
using saved target 'sky'
../../tools/sky/serialdump-linux -b115200 /dev/ttyUSB0
connecting to /dev/ttyUSB0 (115200) [OK]
Rime started with address 45.236
MAC 2d:ec:00:00:00:00 Contiki-2.6-900-ga6227e1 started. Node id is not set
.
CSMA ContikiMAC, channel check rate 8 Hz, radio channel 65491
Starting 'Sensor Acquisition'
Starting Sensor Example.

Temperature=24.81 C (6442)
Humidity=62.27 % (1881)
Light=37.45 lux (92)

```

Figure 3.28: This is a screenshot of the terminal program being executed during the programming of the sensor acquisition .c script to the mote.



The screenshot shows a terminal window with a black title bar and a grey background. The title bar contains three window control icons and the text 'root@instant-contiki: ~/contiki/example'. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The main area of the terminal displays a series of sensor readings, each consisting of three lines: Temperature, Humidity, and Light. Each line includes a numerical value, a unit, and a number in parentheses. The readings are as follows:

Temperature	Humidity	Light
26.01 C (6562)	58.42 % (1754)	35.41 lux (87)
26.04 C (6564)	58.23 % (1748)	221.05 lux (543)
26.06 C (6567)	58.20 % (1747)	239.37 lux (588)
26.12 C (6573)	58.23 % (1748)	256.47 lux (630)
26.16 C (6576)	57.96 % (1739)	254.43 lux (625)
26.16 C (6576)	57.80 % (1734)	155.10 lux (381)
26.20 C (6581)	58.75 % (1765)	114.39 lux (281)
26.25 C (6585)	60.16 % (1811)	61.47 lux (151)
26.26 C (6587)	59.24 % (1781)	35.82 lux (88)
26.27 C (6588)	58.51 % (1757)	35.41 lux (87)
26.29 C (6589)	57.99 % (1740)	35.41 lux (87)
26.30 C (6591)		

Figure 3.29: This is the screenshot of the live readings coming in from the three sensors from the successful programming of the sensor acquisition program.

3.5.5 Using the Cooja Network Simulator

The layout was simulated in the Cooja network simulator. The simulation has been successfully run where the sensor acquisition broadcast program was uploaded to thirteen different motes and they were all talking to each other in a linear fashion. This was due to the derived layout that was designed by the idea that only two in a chain should be able to talk to each other. The data was transferred back and forth successfully to the main central mote. The only problem with this design is that there is not a one way transmission yet, so for example, the main mote still transmits its data out to the four primary motes it is connected to. To fix this, a new type of casting has to be implemented and programmed to the wireless sensor motes. Also the multi-hop network will have to be designed to handle the forwarding of data packets.

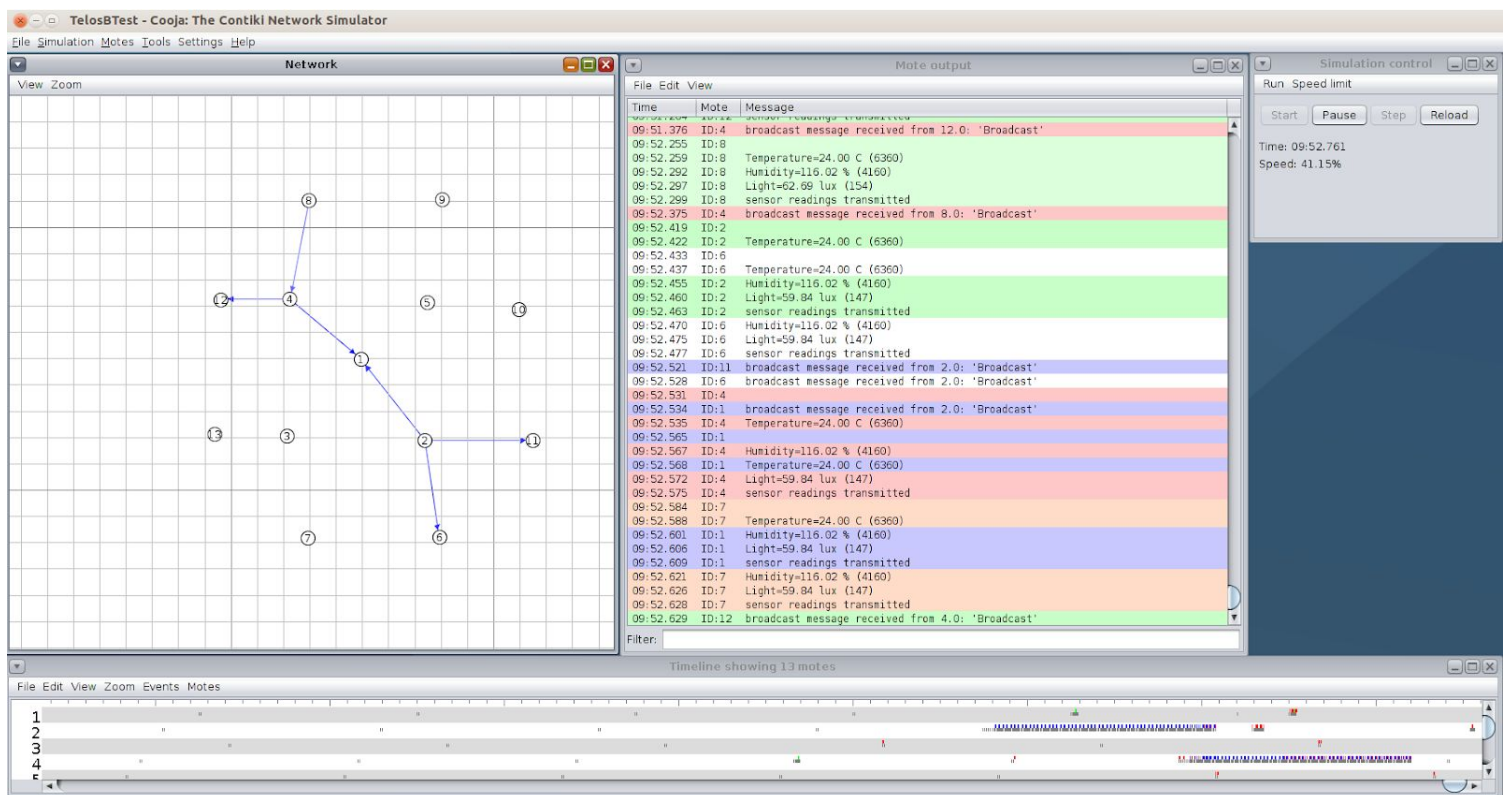


Figure 3.30: This is the Cooja Network simulator running the modified sensor acquisition broadcast program and shows the data being transmitted successfully.

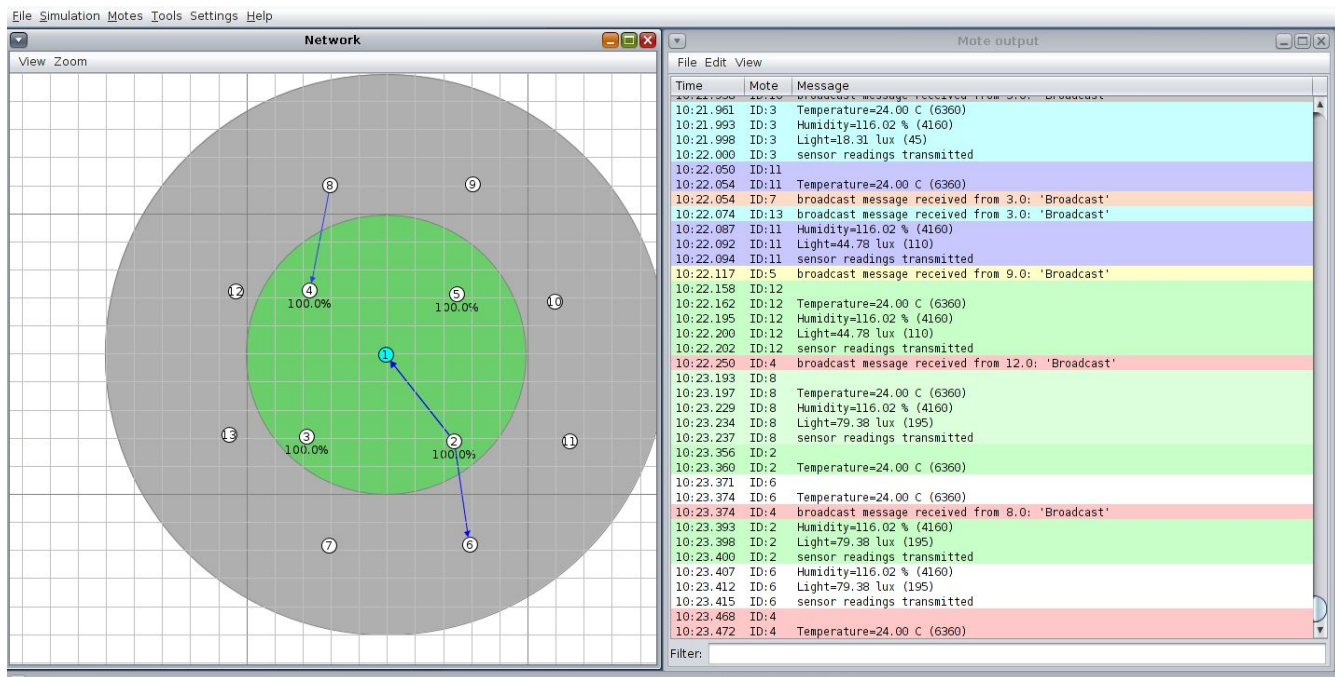


Figure 3.31: This shows the range of the main mote and how it is connected to the four primary motes as the only mote that is connected to more than one mote.

3.5.6 Low Power Modes and Interrupt Processing

The TelosB's on board microcontroller, the MSP430F1611 is capable of going into a reduced power or “sleep” state. There are five of these states and they are all called Low Power Modes. Low Power Mode 0 is where the CPU is disabled along with the MCLK. The MCLK is the CPU clock. Low Power Mode 1 disables the CPU, MCLK, as well as the Digitally Controlled Oscillator DC generator. Low Power Mode 2 disables the CPU, MCLK, as well as the SMCLK. The SMCLK is a high frequency clock used for serial communication and other peripherals. Low Power Mode 3 disables the CPU, MCLK, SMCLK, and the Digitally Controlled Oscillator DC generator as well. Finally, Low Power Mode 4 disables the CPU, MCLK, SMCLK, ACLK, and the Digitally Controlled Oscillator DC generator. The decision was made to use Low Power Mode 3 whenever the sensor mote is not transmitting. This mode is the best for our usage because it turns off everything but a single ACLK. The ACLK is a 32-kHz crystal oscillator. This low frequency clock is great for creating a real time clock that can be used to keep track of time. In our case we would want it to count for ten to fifteen minutes between transmissions. This can be used for interrupt processing.

So the interrupt processing was very simple, it required basic knowledge of interrupts to work with. The Global Interrupt bit had to be set as well as some interrupt enable bits on certain registers in the MSP430. Then the mote was programmed so that every ten minutes, the ACLK would trigger an interrupt to occur and send a transmission. Then it would clear the interrupt bit and go back to Low Power Mode 3. So if the interrupt occurs, the mote wakes up, making the

Low Power 3 mode, the default mode. This means that for most of the time, the mote is in this low power state, meaning much more battery life due to less power consumption.

3.5.7 Type of Casting

The realization occurred that the type of casting that was being performed was highly inefficient. The program that was currently programmed to the motes was broadcasting the data packets to all of the motes in the area. Every neighbor mote in range was receiving these packets of data, whether they needed to or not. This is not a very good design, but it is great for testing purposes. The final design needs to include either a unicast or an anycast.

The unicast is a type of casting that unlike broadcast, can be sent to a specific neighbor with the destination address in mind. However, the catch is that this “neighbor” has to be exactly that, a neighbor. If the mote you wish to specifically transmit to is out of range, then you cannot talk to it. There is no forwarding for this type of casting. The other type similar to this is a multicast where the mote transmits to a specific set of multiple neighbors, but this suffers from the same problem as before where if the destination node is not in range, the data will never make it to where it is supposed to. There are two ways to handle this. Either three separate programs will need to be made to handle the three different types of nodes (exterior, mid-range, and base unit), or anycast can be utilized. The anycast method basically finds the path to its destination by travelling through its neighbors with the shortest distance to the end destination to finally reach it. This works well when there are multiple routes that the data can take to get to the base unit. In this case we would have to abandon the single route method which is a foreseeable option to fix the routing issue.

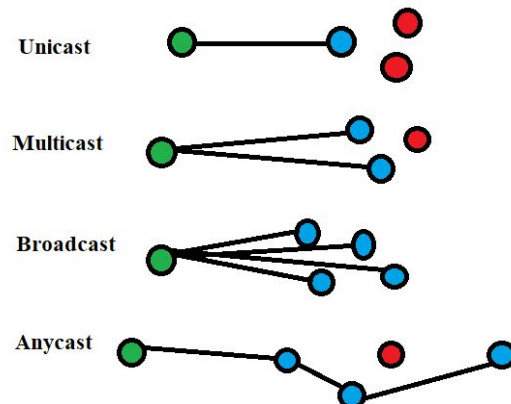


Figure 3.32: This drawing visually shows the different types of casting.

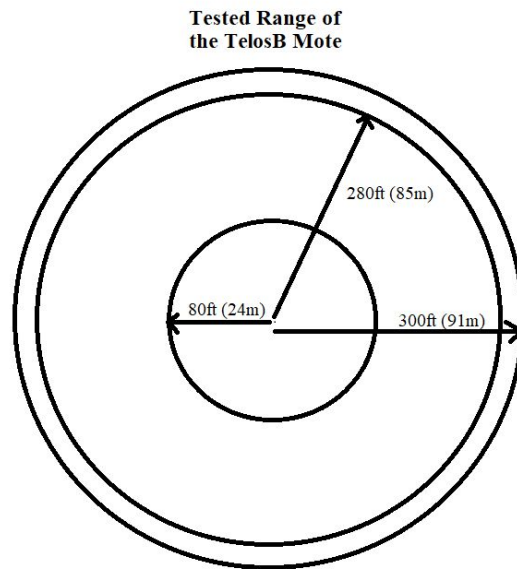


Figure 3.33: This drawing visually shows the different ranges of the TelosB mote.

3.5.8 Multi-Hop Future Design Plan

The design for the future needs to take advantage of multi-hop routing. This style of routing is used for the end goal of forwarding the data to a destination node that is outside the range of the sender node. There are two main methods for this type of connection. The first is called “flooding”. Flooding consists of a lazy style of communication that is very inefficient. Basically what occurs is that the node with the data that needs to be sent transmits it to all of its neighbors, no matter who they are. So this is a basic broadcast to everyone. Then each of the nodes that receives the data also broadcasts it to every single one of its neighbors. After all of these propagating waves of signal being sent, at some point it will reach the destination. However, this is a huge waste of power and time, and would cause way too many broadcasts to occur. The other option which is much more reliable is called the shortest path algorithm. This basically is an algorithm used to calculate the shortest path cost based on distance and traffic factors. There is a certain cost for each route the packet could take and it figures out which cost ends up being the least expensive and opts for that route. The algorithm would have to be implemented with the new team of people joining this project during the summer.

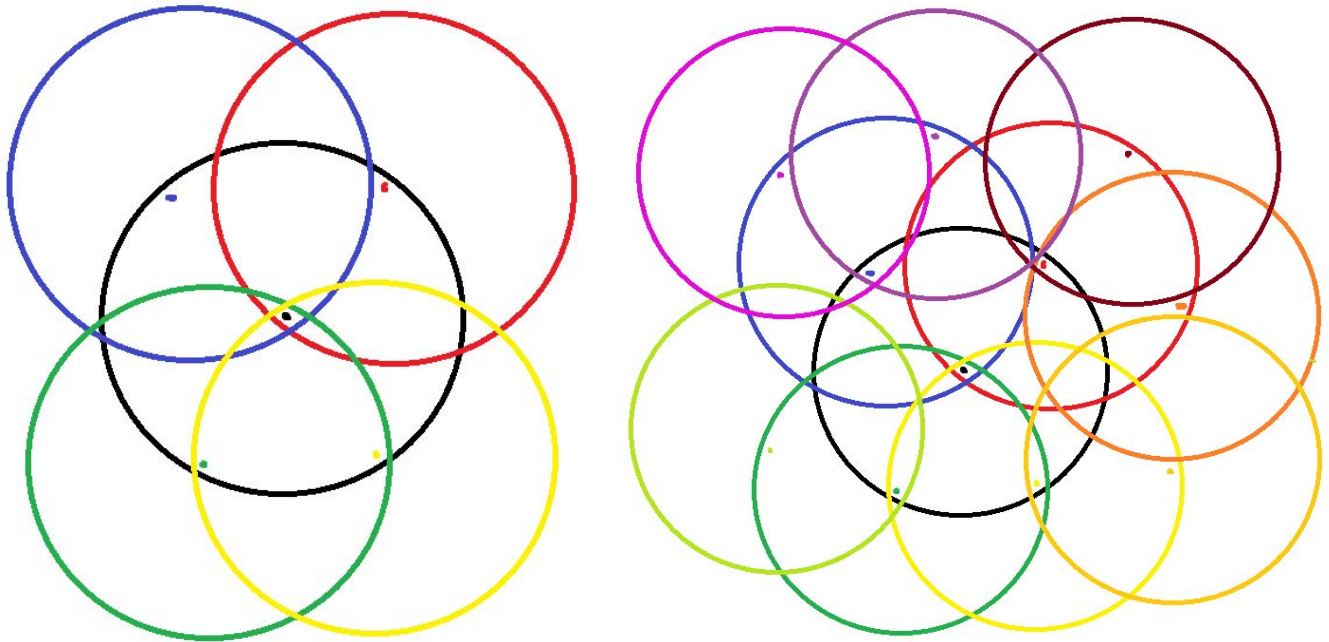


Figure 3.34: This drawing illustrates the complexity added per layer of nodes.

3.5.9 Design with Energy Harvesting

Once the energy harvester is built, we will connect the alternate sources' power lines to the actual mote itself. The AA battery pack that was attached to the underside of the mote has been desoldered to make room for the energy harvester. The energy harvester will have to supply its collected power to the load to supplement the TelosB while it loses energy throughout its regular usage. The mote has a very low power mode that it will go into in between measurements that will hopefully minimize the power consumption, allowing the energy harvester to have a chance to continuously "refill" the power. This is the main idea behind the project so it has to work somewhat efficiently. Along with the MPPT calculations and the continuous collection of solar during the day, it should be enough to supplement the power during the motes operation.

3.6 IEEE Standards

For wireless communications, it is important that devices operate within the appropriate frequency band as specified by IEEE standard 802.15.4. The TelosB sensor mote was verified to be operating within an appropriate frequency, 2.4 GHz. The sensor fits the purpose of this standard which is directed towards "ultra low complexity, ultra low cost, ultra low power consumption, and low data rate wireless connectivity among inexpensive devices. The raw

data rate is high enough (250 kb/s) to satisfy a set of applications but is also scalable down to the needs of sensor and automation needs (20 kb/s or below) for wireless communications. [18]” The sensor must continually operate within 2400 to 2483.5 MHz, as well as hold a transmission range of at least 10 meters. The mote can transmit 30 meters indoor and roughly 95 meters outside. There is also no network layer defined within the standard so routing may be an issue. The other main takeaway from this engineering standard is that there are only two types of nodes recognized. The first is a FFD(Full Function Device, which can act as a coordinator to send and receive packets of data. The second is a RFD(Reduced Function Device), which can only communicate with a FFD and not any other RFD’s. It is a simple node that does not have any advanced control capabilities in the wireless sensor network.

Regarding the standalone energy harvesting network, there is not as strict regulation considering that it will not be connected to the power grid. However, IEEE standard 1526 provides recommendations for the evaluation of stand-alone photovoltaic systems [19]. The converter and the panel were hooked and verified to be working per the time recommended by the standards; however, tests did not yet include battery charging or closed loop control. Based on the IEEE standard, the circuit used in this experiment would also require additional protection measures. This includes a diode in series with the solar panel to avoid reverse current into the solar panel. The reverse current from the battery however would not be an issue due to the second diode of the converter already serving this purpose. Also, additional circuitry would be added for appropriate switching to disconnect the battery at the end of its charge cycle.

4. FINAL PRODUCT DETAILS

4.1 Results

The converter was created (Figure 3.17) and tested in buck and boost mode using a 5.06V input voltage, a 33Ω load resistor, and a function generator which provided a variable duty cycle PWM signal for the MOSFETs. The converter appears to perform the best in buck mode, but also performs well in boost mode.

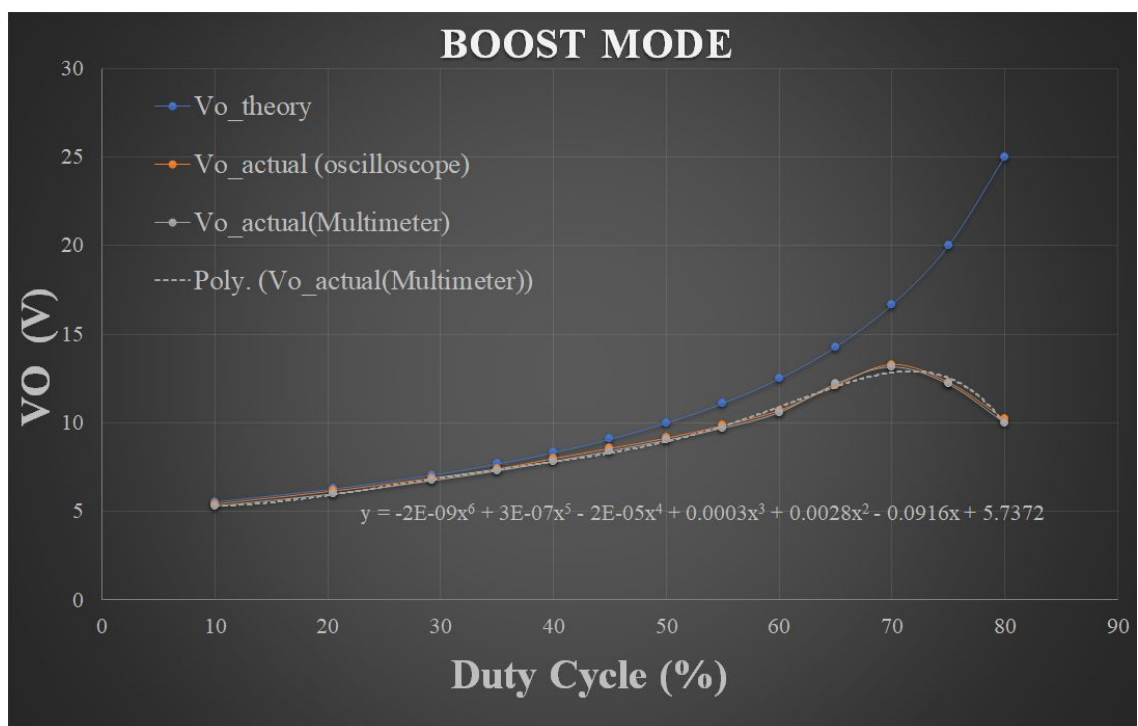


Figure 4.1: The results of the DC/DC converter in boost mode.

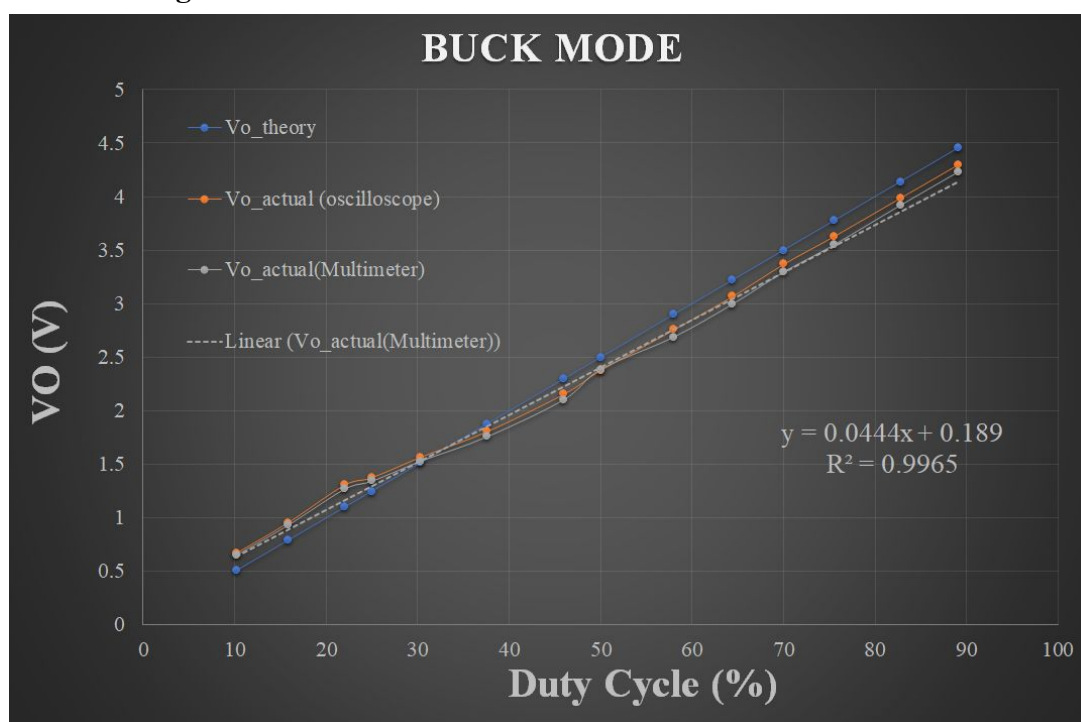


Figure 4.2: The test results of the DC/DC converter in buck mode.

Boost Mode ($V_{in} = 5V$, $RL = 33.1 \Omega$)				
% Duty	Vo_Theory (V)	Vo_Actual (Oscilloscope)	Vo_Actual (Multimeter)	% error
10	5.556	5.44	5.3	2.088
20.5	6.289	6.18	6	1.733
29.2	7.062	6.9	6.74	2.294
35	7.692	7.41	7.29	3.666
40	8.333	8	7.83	3.996
45	9.091	8.57	8.41	5.731
50	10	9.15	9.03	8.5
55	11.111	9.85	9.71	11.349
60	12.5	10.7	10.59	14.4
65	14.286	12.1	12.21	15.302
70	16.667	13.3	13.17	20.202
75	20	12.3	12.2	38.5
80	25	10.2	10	59.2

Table 4.1: Test results for boost mode, tabulated.

Buck Mode ($V_{in} = 5 V$, $RL = 33.1 \Omega$)				
Duty (%)	Vo_Theory(V)	Vo_Actual (oscilloscope)	Vo_Actual (Multimeter)	% error
10.2	0.51	0.676	0.655	32.549
15.8	0.79	0.958	0.932	21.266
22	1.1	1.31	1.27	19.091
25	1.25	1.38	1.345	10.4
30.3	1.515	1.57	1.53	3.63
37.6	1.88	1.81	1.76	3.723
46	2.3	2.16	2.107	6.087
50	2.5	2.38	2.388	4.8
58	2.9	2.76	2.69	4.828
64.4	3.22	3.07	2.996	4.658
70	3.5	3.37	3.295	3.714
75.5	3.775	3.63	3.55	3.841
82.8	4.14	3.99	3.92	3.623
89.1	4.455	4.3	4.23	3.479

Table 4.2: Test results for boost mode, tabulated.

4.2 Proposal for Future Work

Considering the small size of a wireless sensor, a small wind turbine can be added to the energy harvesting network. Turbines do best in the outdoors or in air vents; however, in areas such as Fresno, wind speeds do not typically go beyond 5 mph. However, if a wind sensor is used within a vent, it could more readily rely on power supplied by the wind turbine. Piezoelectric harvesting is the conversion of mechanical strain, such as vibrations, into electrical energy. There is a way to extend the harvesting of wind energy by using either a piezoelectric strip or a piezoelectric fan. In this project, the piezoelectric strip would be preferred over the wind turbine given its small size and low cut-in voltage. Depending on the sensor application, different types of energy sources become viable.

In addition, to charge a lithium ion battery, it is recommended to implement an I2C battery monitor to interface with the microcontroller. This would increase the processing speed of the microcontroller since it would not have to undergo as many ADC conversions and reduce the network's size. A digital compensation technique should also be considered in the future to complete a closed loop implementation.

With regards to the Wireless Sensor Network, the multi-hop shortest route algorithm should be implemented as well as the anycast procedure. This would make the future design much cleaner and easier to work with, as well as far more efficient.

4.3 Budget

The following table shows the various hardware items that have been purchased along with their price and availability. The funding for the project was provided by the research funding from the National Science Foundation grant awarded to both advisors of this project. Any additional costs were covered by the team members of this project.

Item	Supplier	Cost
TelosB Sensor Mote(3)	Crossbow Technologies	\$100
Monocrystalline Solar Panel	Adafruit	\$40
Li-Ion Battery	AA-Portable Power Co.	\$7
Passive Components	Digi-Key	\$15
PIC16F1778	Mouser	\$7
19TQ015 Schottky Diodes	Mouser	\$10
MAX4172 Current Sensor	Mouser	\$6
LT6005 Op Amps	Mouser	\$3
19TQ015 Schottky Diode	Mouser	\$5

5. LOGISTICS

5.1 Division of Labor

Task	Team Member
Energy Harvesting Theory	Luis/Jasmine
Wireless Sensor Theory	Jared
Solar Devices	Luis/Jasmine
Sensor Operation	Jared
Sensor Mote Programming	Jared
Interrupt Processing + Networking	Jared
DC-DC Converter Design	Luis/Jasmine
Simulations	Luis/Jasmine
Energy Storage	Luis/Jasmine

5.2 Work Completed

A successful simulation of the controller was implemented to produce constant current and constant voltage. A successful prototype for the two-switch buck-boost converter has also been created, but still requires the implementation of closed loop control. The microcontroller is capable of switching between converter modes and applying precise duty cycles. The wireless sensor mote for this project has been verified to be capable of sensing accurate changes to temperature and light intensity as well as being able to communicate with other motes for the development of the wireless sensor network. The wireless sensor network itself has not been established however, but the sensors were found to be able to obtain data in real time, as well as broadcast to all other motes within a 300 foot range given no obstacles.

6. STATEMENT ABOUT TEAM MEMBERS

Jared Sarajian:

Relevant courses completed include: Digital Logic, Microprocessor Architecture and Programming, Software Engineering, Signals and Systems, Power Electronics, CAE in Digital Design and Advanced Computer Architecture.

Relevant courses in progress include: Probabilistic Engr. Sys. Analysis, Wireless Networking, Digital Signal Processing, and Advanced Circuit Analysis

Special Skills include:

- Basic programming knowledge in: Java, C++, Verilog, Assembly, Matlab, HTML5, and Python

Luis Ortiga:

Relevant courses completed include: Switching Theory & Logic Design, Electromagnetics & Maxwell's Eq., Electromechanical Systems & Energy Conversion, Digital Logic, Advanced Circuit Analysis, Microprocessor Architecture & Programming, Electronics II

Relevant courses in progress include: Power Electronics

Special skills include:

- Python/C++ programming, PIC and Verilog programming, Matlab, Assembly

Jasmine Lopez:

Relevant courses completed include: Electromechanical Systems and Energy Conversion, Electrical Power Systems, Digital Logic, Signals and Systems, Microprocessor Architecture and Programming, Electronics II, Control Systems, Power Electronics

Relevant courses in progress include: Programmable Logic Controllers ECE 119L

Work experience relative to engineering: Power systems internship

Special skills include:

- Assembly, C programming, Matlab

References

- [1] Yao, Chuan & Ruan, Xinbo & Weijie, Cao & Chen, Peilin. (2014). A Two-Mode Control Scheme With Input Voltage Feed-Forward for the Two-Switch Buck-Boost DC–DC Converter. *Power Electronics, IEEE Transactions on*. 29. 2037-2048. 10.1109/TPEL.2013.2270014.
- [2] H. Dean Venable. (1983) *The K Factor: A New Mathematical Tool for Stability Analysis and Synthesis*. Venable Industries, Inc. Rancho Palos Verdes, California.
- [3] Mihnea Rosu-Hamzescu, Sergiu Oprea. “AN1521 - Implementing Solar Panel MPPT Algorithms.” *Microchip*. Microchip Technology Inc.
Available:<http://ww1.microchip.com/downloads/en/AppNotes/00001521A.pdf>
- [4] “AN2321 - Solar MPPT Battery Charger for the Rural Electrification System.” *Microchip*. Microchip Technology. Available:
<http://ww1.microchip.com/downloads/en/AppNotes/00002321A.pdf>
- [5] “Digital control strategy for a buck converter operating as a battery charger for stand-alone photovoltaic systems” *Science Direct*. Available:
<https://www.sciencedirect.com/science/article/pii/S0038092X16305217>
- [6] “Demystifying Type II and Type III compensators.” Texas Instruments.
<http://www.ti.com/lit/an/slva662/slva662.pdf>
- [7] S. Keeping. “Managing Converter In-rush Current.” *Digikey.com*
<https://www.digikey.com/en/articles/techzone/2012/jun/managing-converter-in-rush-current>
- [8] Li-Ion Battery Spec-sheet. *Batteryspace.com*
<https://www.batteryspace.com/prod-specs/1254.pdf>
- [9] Design Considerations for DC-DC Converters in Fast-Input Slew Rate Applications
<http://www.ti.com/lit/an/slva693/slva693.pdf>
- [10] A. Fayed. “An Overview of the Fundamentals of Battery Chargers.” *IEEE.org*.
<https://ieeexplore.ieee.org/document/7870048>
- [11] H. Yau, etc. “Maximum power point tracking and optimal Li-ion battery charging control for photovoltaic charging system.” *ScienceDirect*.
<https://www-sciencedirect-com.htmlproxy.lib.csufresno.edu/science/article/pii/S0898122111011047>
- [12] Lee, Y. (2009, April). A Compensation Technique for Smooth Transitions in a Noninverting Buck–Boost Converter. Retrieved March 26, 2019, from
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4770211>

- [13] Documentation on Programming with Contiki
http://anrg.usc.edu/contiki/index.php/Build_your_own_application_in_Contiki
- [14] Contiki Main Site
<http://www.contiki-os.org/start.html>
- [15] Where to Access all of the datasheets for the mote (datasheets all referenced)
<https://www.advanticsys.com/shop/mtmcm5000msp-p-14.html>
- [16] M. Murnane and A. Ghazel. "A Closer Look at State of Charge (SOC) and State of Health (SOH) Estimation Techniques for Batteries." Semantic scholar.org
<https://pdfs.semanticscholar.org/964c/93c82bf2ee272c6491b3cb85a06529e935d2.pdf>
- [17] "Bypass Diodes in Solar Panels". Electronics-tutorials.ws
<https://www.electronics-tutorials.ws/diode/bypass-diodes.html>
- [18] IEEE Computer Society. IEEE Std 802.15.4™-2011. *IEEE*. Available:
<https://ieeexplore-ieee-org.hmlproxy.lib.csufresno.edu/stamp/stamp.jsp?tp=&arnumber=6012487>
- [19] IEEE Standards Coordinating Committee 21. IEEE Std 1526™-2003. *IEEE*. Available:
<https://ieeexplore-ieee-org.hmlproxy.lib.csufresno.edu/stamp/stamp.jsp?tp=&arnumber=1263344>

APPENDIX A

Power Dissipation of B Internal Voltage Regulator

Thermal Considerations

The MIC5207 is designed to provide 180mA of continuous current in a very small package. Maximum power dissipation can be calculated based on the output current and the voltage drop across the part. To determine the maximum power dissipation of the package, use the junction-to-ambient thermal resistance of the device and the following basic equation:

$$P_{D(MAX)} = \frac{(T_{J(MAX)} - T_A)}{\theta_{JA}}$$

$T_{J(max)}$ is the maximum junction temperature of the die, 125°C, and T_A is the ambient operating temperature. θ_{JA} is layout dependent; Table 1 shows examples of junction-to-ambient thermal resistance for the MIC5207.

Package	θ_{JA} Recommended Minimum Footprint	θ_{JA} 1" Square Copper Clad	θ_{JC}
SOT-23-5 (M5)	235°C/W	170°C/W	130°C/W

Table 1. SOT-23-5 Thermal Resistance

The actual power dissipation of the regulator circuit can be determined using the equation:

$$P_D = (V_{IN} - V_{OUT})I_{OUT} + V_{IN} I_{GND}$$

Substituting $P_{D(max)}$ for P_D and solving for the operating conditions that are critical to the application will give the maximum operating conditions for the regulator circuit. For example, when operating the MIC5207-3.3BM5 at room temperature with a minimum footprint layout, the maximum input voltage for a set output current can be determined as follows:

$$P_{D(MAX)} = \frac{125^\circ\text{C} - 25^\circ\text{C}}{235}$$

$$P_{D(MAX)} = 425\text{mW}$$

The junction-to-ambient thermal resistance for the minimum footprint is 220°C/W, from Table 1. The maximum power dissipation must not be exceeded for proper operation. Using the output voltage of 3.3V and an output current of 150mA, the maximum input voltage can be determined. From the Electrical Characteristics table, the maximum ground current for 150mA output current is 3000μA or 3mA.

$$455\text{mW} = (V_{IN} - 3.3)150\text{mA} + V_{IN} \cdot 3\text{mA}$$

$$455\text{mW} = V_{IN} \cdot 150\text{mA} - 495\text{mW} + V_{IN} \cdot 3\text{mA}$$

$$920\text{mW} = V_{IN} \cdot 153\text{mA}$$

APPENDIX B

Power Consumption of the Transmitting



CM5000 DATASHEET

- Battery monitor
- QLP-48 package, 7x7 mm
- Complies with ETSI EN 300 328, EN 300 440 class 2, FCC CFR-47 part 15 and ARIB STD-T66
- Powerful and flexible development tools available

The TX power consumption of the CC2420 is dependent on the programmed output power:

Output Power [dBm]	Current Consumption [mA]
0	17.4
-1	16.5
-3	15.2
-5	13.9
-7	12.5
-10	11.2
-15	9.9
-25	8.5

Figure 11: Current Consumption vs. Output Power

In low power applications, the CC2420 should be powered down when not being active. Extremely low power consumption may be achieved when disabling also the voltage regulator, but this will require reprogramming of the register and RAM configuration.

The CM5000 mote has two antenna options—and internal antenna built into the module and an external SMA connector for connecting to external antennas. By default, CM5000 is shipped with the internal antenna enabled. If an application requires an external antenna or a different directional pattern than the internal antenna, an SMA connector may be installed and an antenna may be connected directly to CM5000's SMA female connector.

APPENDIX C

Various Mote Characteristics and Chip Pinouts

OPTIONS FOR FREQUENCY ASSIGNMENTS			
Geographical regions	Europe	Americas	Worldwide
Frequency assignment	868 to 868.6 MHz	902 to 928 MHz	2.4 to 2.4835 GHz
Number of channels	1	10	16
Channel bandwidth	600 kHz	2 MHz	5 MHz
Symbol rate	20 ksymbols/s	40 ksymbols/s	62.5 ksymbols/s
Data rate	20 kbits/s	40 kbits/s	250 kbits/s
Modulation	BPSK	BPSK	Q-QPSK

