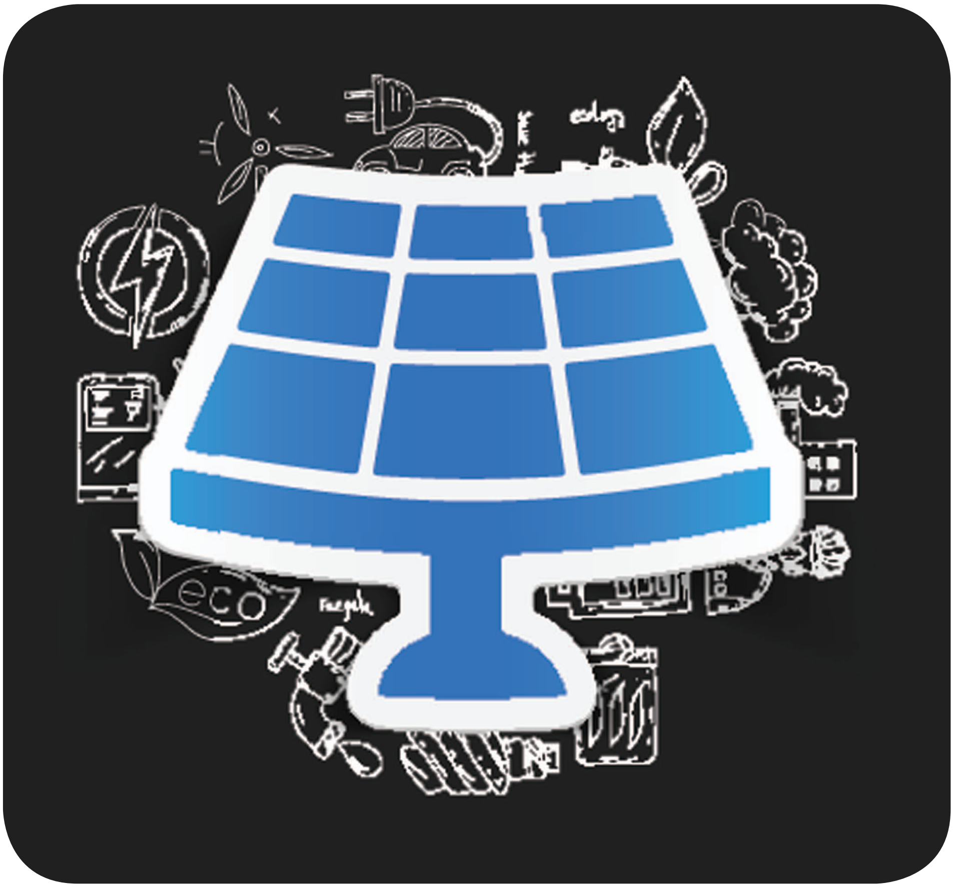
Maximum power point tracker: A senior design experience

By A. Buke Hiziroglu, Macy Payne, Michael Glembotzki, and Mark G. Thompson



The academic calendar at Kettering University in Flint, Michigan, is based on quarters, with each term having 10½ weeks of lecture time. The electrical engineering senior design course at Kettering is only a quarter-term- long project, and the projects vary each term depending on the professor. The course is structured mainly for three-student teams to work independently, and interaction with the professor is mostly limited to checking each team's progress during the milestones.

The project we were given was a maximum power point tracker (MPPT) for a photovoltaic system. After being provided with basic components including an Arduino microcontroller and a gate driver, a budget of US$106 was allotted for each team. To satisfy each milestone in the senior design course, a technical timeline was established in the beginning of the term, as presented in Table 1.

System description

An MPPT typically consists of a microcontroller, a power converter, and a load to receive the maximum possible power from photovoltaic solar panels. The purpose of the MPPT is to sample the output voltage and current of the photovoltaic cells to obtain the maximum power for any given weather condition. The goal of this project is to drive the Shurflo water pump in conjunction with the MPPT to pump water to a prespecified elevation based on the buck converter topology.

Photovoltaic solar cells are inherently complex devices since their current-voltage (I–V) characteristic offers a nonlinear behavior. The solar irradiance strength, the type of solar cell, and the solar panel temperature influence the power conversion directly. The I-V characteristics, (Fig. 1) of the Arco Solar/Siemens M75 photovoltaic cell used in the validation of the MPPT was regenerated from the

Digital Object Identifier 10.1109/MPOT.2015.2429411
Date of publication: 14 November 2016

TABLE 1. The technical timeline.

TASK	DEADLINE	POINTS POSSIBLE
Milestone 1: Studying photovoltaic power generation	Week 2 Monday	10
Milestone 2: Studying the design options	Week 3 Monday	10
Milestone 3: Power circuit design and prototyping	Week 5 Monday	10
Milestone 4: MPPT control design	Week 7 Monday	10
Milestone 5: Control algorithm	Week 8 Monday	10
Milestone 6: Hardware implementation	Week 9 Monday	10
Milestone 7: Produce a functioning product	Week 10 Monday	10
Milestone 8: Final demonstration	Week 10 Wednesday	30

data found in the Arco Solar Inc. installation guide. Stronger irradiance increases the magnitude of current while keeping the voltage at the same level. Although temperature rise may lower the voltage, the current remains at nearly the same level.

Using the current and voltage values from Fig. 1, the power curves ($P = VI$) were obtained with each curve having only a single best operating point. The best operating point defines the maximum power point for various power curves, as shown in Fig. 2. A controlled converter is necessary to maintain operation at the maximum power point.

To determine the I–V characteristic of the Shurflo RV water pump, an experiment was conducted during Milestone 2. In this experiment, the voltage applied to the dc motor of the pump is adjusted to have 200-mA increments on current up to a level when the water height becomes 1 ft. Then, the voltage is reduced with 200-mA steps on current.

As can be seen in Fig. 3, the I–V characteristic of the pump indicates a hysteresis behavior around 1.4 A, which is important for piloting it depending on if the pump is just starting or has been already running. Also observed from Fig. 3 is that once the water pump operates at a current level above 1 A, its I–V characteristic presents an almost linear behavior. From the current-voltage characteristic data, the input resistance of the Shurflo RV water pump was calculated as approximately 5 Ω when water is being pumped to a height of 1 ft.

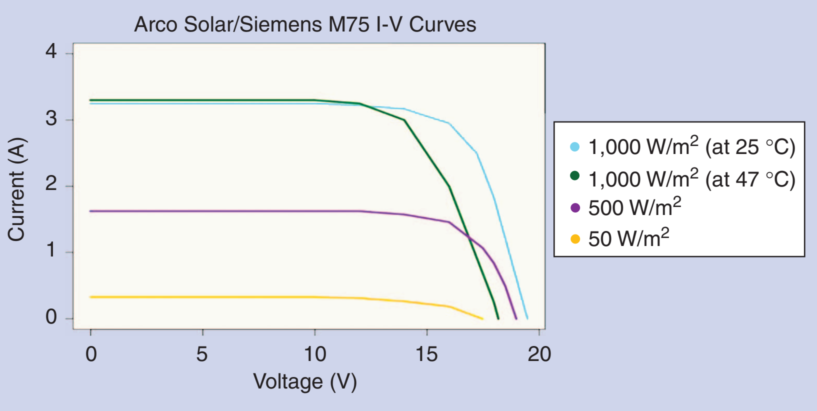


FIG1 The current-voltage characteristics regenerated from the Arco Solar/Siemens M75 solar panel data.

MPPT hardware development

Buck converter design

A buck converter is a dc-to-dc converter, which steps down the input voltage to a lower level with a topology that utilizes a transistor, a diode, an inductor, a capacitor, and a load. The initial critical value of the inductance was calculated as 25.39 μH for a continuous inductor current at a switching frequency of 40 kHz, assuming that the components were ideal. However, in the actual design, an inductor with an inductance of 1 mH was selected to account for nonideal properties of the components along with a switching frequency of 55 kHz. To minimize the ripple at the output voltage of the buck converter when the duty cycle was increased to 80%, a filtering capacitor was necessary across the output terminals. The capacitance of this capacitor was calculated with the following parameters for the buck converter,

$$L = 1\,\text{mH} \quad f = 55\,\text{kHz}$$
$$D = 80\% \quad \frac{\Delta V_o}{V_o} = 2\%$$
$$C = \frac{1-D}{8L\left(\frac{\Delta V_o}{V_o}\right)f^2}$$
$$= \frac{0.2}{8(1\times10^{-3})(0.02)(60\times10^{3})^2}$$
$$= 0.35\ \mu F.$$

For a better ripple performance, the 0.35 μF capacitor was raised to 0.5 μF. In the implementation, two 1-μF capacitors were connected in series to get 0.5 μF.

To verify the performance of the designed buck converter, a simulation was conducted using PSpice. In the simulation, a 10-V dc power supply was connected to the drain of the metal–oxide–semiconductor field-effect transistor. At 80% duty cycle, the simulation yielded an output voltage of approximately 8.3 V, as can be observed in Fig. 4, where the green signal represents the output voltage.

Figure 5 shows when the buck converter is constructed with these parameters, where the signal applied to the gate is shown in yellow and the experimental voltage output voltage is in green. The output voltage was approximately 8 V, and oscillations on the output voltage were reduced significantly with a ripple voltage considerably lower than 2% at 80% duty cycle.

With the observations in Figs. 4 and 5, it can be concluded that the simulation results and the experimental results were in agreement.

The output waveforms for 20%, 50%, and 75% duty cycles were also analyzed when the 0.5-μF capacitance was placed at the output of the buck converter. The output voltages were consistent with the corresponding duty cycles similar to the waveform in Fig.5. Finally, to further improve the performance of the buck converter, the fast recovery 4002 diode of the buck converter was replaced with a Schottky diode, since the pump as the actual load of the buck converter demands high power.

Current sensor and analog filter

The M75 solar panel is capable of producing a maximum current of 3 A. The amount of current through the buck converter must be sensed so that the microcontroller can react according to the power curve data for the M75 solar panel. The current sensing was accomplished by means of a low-resistance resistor of 5 mΩ to minimize the amount of voltage drop across the resistor. The current-sensing resistor was connected from the positive terminal of the solar panel to the positive terminal of the buck converter input. If the solar panel were to reach its maximum current output of 3 A, it is expected from Ohm's law that there would be 0.015 V across this resistor. Since this is a very small voltage, it does not provide the resolution needed to truly control the system.

Therefore, across this resistor there is an amplifier with a gain of 100 to amplify this voltage. This means that we would expect a maximum of 1.5 V on the output side of the current sensing amplifier, which is well below the rated 3.3-V maximum that the microcontroller can handle without damage, according to Arduino's data. Since this output is going to be read by the microcontroller, it is necessary to reduce the noise in the system. Therefore, a 0.1-μF capacitor was connected as

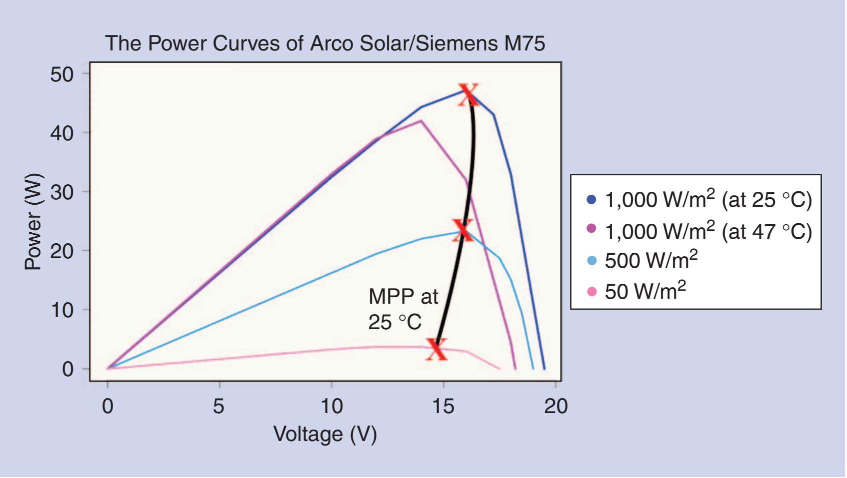


FIG2 The power curves for the Arco Solar/Siemens M75 solar panel.

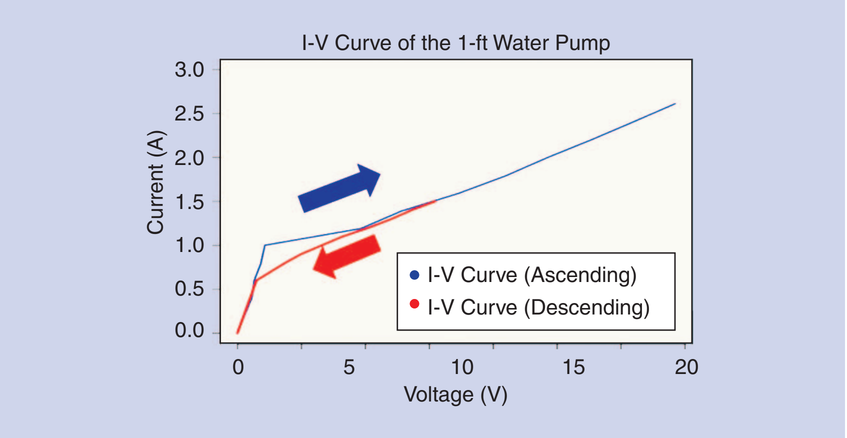


FIG3 The current-voltage characteristic of the water pump when pushing water to a height of 1 ft.

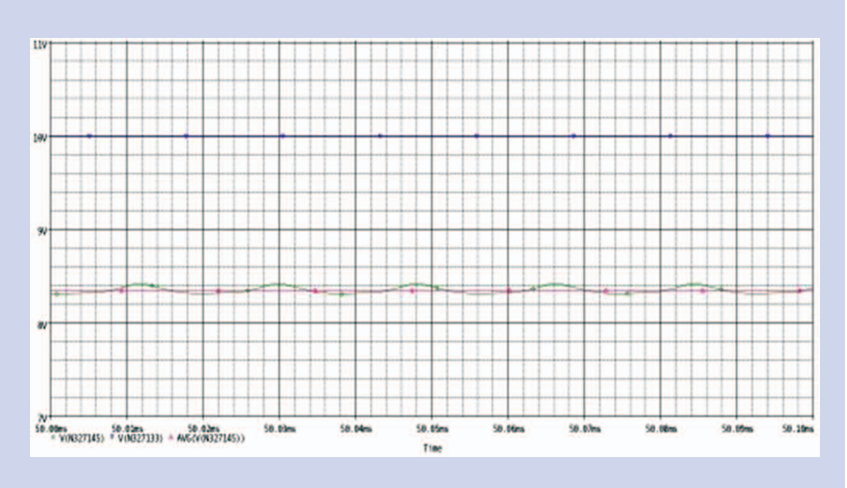


FIG4 The PSpice simulation output for the buck converter at 80% duty cycle, with the input voltage (blue), output voltage (green), and the average of the output voltage (pink).

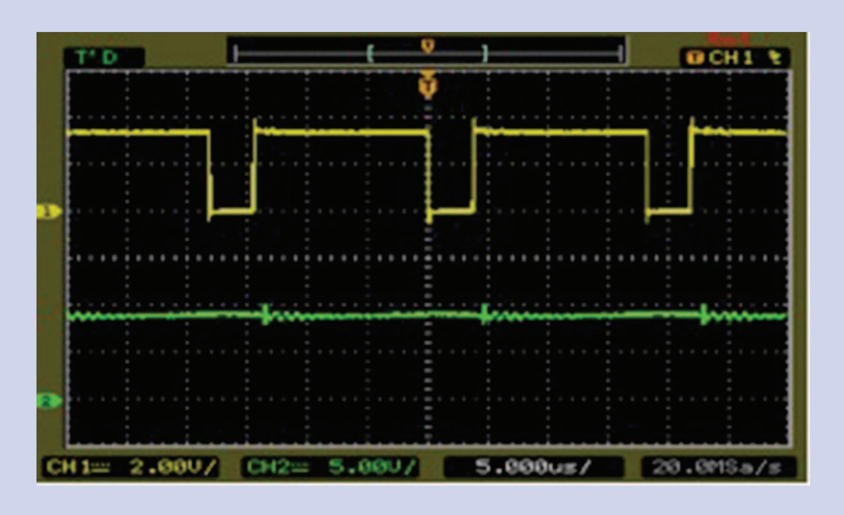


FIG5 The input gate signal (yellow) and the output voltage of the buck converter (green) with a duty cycle of 80%.

an analog filter between the output pin of the amplifier and the ground to reduce the noise.

The voltage sensor and analog filter

Also connected to the junction of the current sensing resistor and the buck converter input is the voltage sensor. It is known that the maximum voltage the Arduino can handle, as previously mentioned, is 3.3 V, while the solar panel can produce 19.2 V at the maximum, as stated by Arco Solar Inc. Therefore, a voltage divider with a low-voltage arm fewer than 3.3 V is necessary.

Another consideration is that the resistors in the sensor circuit must be fairly large to avoid large current flow from the system. Therefore, the high-voltage arm resistor is considered to be 100 kΩ. Without a safety factor, this means that the low-voltage arm resistor would be just less than 21 kΩ. However, in providing a safety factor, it was necessary to maintain as high a resolution as possible. Therefore, it was decided to limit the safety factor to 5%. This means that at 19.2 V, the output of the voltage divider would be approximately 3.13 V, which is under the maximum voltage of 3.3 V for the Arduino. This also means that if the solar panel was able to produce up to 20 V, the Arduino would not be damaged because, at 20 V, the output to the Arduino would be 3.264 V, which is still less than its maximum rating.

To meet these specifications, a 19.5-kΩ resistor on the low side of the voltage divider was necessary. Two 39-kΩ resistors were put in parallel on the low-voltage arm to achieve the goal. In addition to the resistors of the voltage divider, a 0.01-μF capacitor was added as an analog filter in the low-voltage arm to make the voltage sensing more accurate by reducing the noise content in the voltage sensor.

Voltage regulator

Arduino requires a power supply with a voltage between 7 V and 12 V for its operation. As the replacement of a 9-V battery could be an inconvenience in the field, it was decided to use the solar panel to drive the Arduino. However, it is not possible to connect the solar panel directly to the V_{in} pin of the Arduino. Therefore, a voltage regulator needs to be connected between the two devices. The V_{in} pin can be used as a port to power the board, but it must receive a voltage above 7 V to maintain stability. Also, this pin should not receive more than 12 V so that the on-board regulator will not overheat and cause damage to the board and the microcontroller.

In this project, an external 12-V regulator was implemented. A set of data was collected to verify that the voltage regulator never exceeded 12 V within the expected operating conditions. The maximum voltage output was 11.89 V, thus it was considered safe for the Arduino. Using this regulator, the Arduino turns on when the output side is at just over 4 V, where the input is approximately 5 V, and this is still below where the buck converter begins to operate (when the input is approximately 8 V). As a result, this regulator was deemed an acceptable solution, and it was implemented.

MPPT software development

To achieve the maximum power point, the output voltage of the buck converter is controlled by the duty cycle of the gate signal. The duty cycle is selected for the Arduino based on Yu and Lin's hill-climbing algorithm. Since this method does not depend on many computations, the program requires less memory and results in a faster operation compared to other algorithms. The input voltage and current are read in via *analogRead()* and multiplied to get the average power. A decision is made by the microcontroller based on the comparison of the present and previous values of the power, whether the duty cycle can be increased or decreased by $\Delta D = 0.5\%$. The starting duty cycle value is set to 50% of the period.

Based on our observations, one drawback of this method is that the controller never lands on the maximum power point but instead oscillates between just under and just over the maximum power point. Also, this method is susceptible to being thrown off by fast changes

To achieve the maximum power point, the output voltage of the buck converter is controlled by the duty cycle of the gate signal.

in input power levels and will take some time to recover again. The hill-climbing algorithm of the MPPT system is presented in the flowchart in Fig. 6.

Arduino's Due board was used as the microcontroller for this project. The Arduino Due has a master clock speed of 84 MHz with 54 digital input/output pins, in which 12 are used for pulse-width modulation (PWM) generation, according to Arduino's data. To get the right frequency, which is based on our inductor and capacitor calculations, we had to configure the PWM channel. Convenient PWM channels are on the digital output pins 6, 7, 8, and 9. We decided to use pin 6, which is configured to PWM channel 7 and uses the internal clock *B*, according to Arduino's data. The buck converter begins operating around a dc input of 8 V. To show the voltage level where the microcontroller begins running through the program, the light-emitting diode (LED) "L" on the Arduino Due glows green.

Digital filter programming in the form of a moving average was added for each sensor. The program reads values from the voltage and current sensors, where each sensor has its own array (index) and stores the sensor readings. Each time a value is stored, the program moves to the next location in the index until it reaches a set number of samples used in calculating the total. In this case, it has been set up to store the last ten values in this index and averages them to find an approximation of the voltage and current during a period of time. Once ten values are stored, then the index is returned to the location of the oldest reading and stores a new one in its place.

The program then continues to loop through the index in this manner. One important aspect of the moving average is that the variables *voltageavg*, *currentavg*, *total*, and *total2* were made to be "float" variables instead of "int" values in the program. Without using "float" values, a sensor reading with a decimal will be rounded to the next integer, which greatly reduces the resolution of the reading. Therefore, to allow decimals to be included in the calculations using sensor values for the control algorithm, "float" values were necessary.

The functional test yielded that our system worked perfectly under constant current conditions as well as constant voltage conditions.

MPPT final system details

MPPT system schematic

Figure 7 represents the schematic for the complete MPPT system designed and implemented for this project. It includes the photovoltaic solar panel, the buck converter, its gate driver circuit, voltage and current sensor circuits, and the water pump. The circuit schematic also highlights the connections to the microcontroller at DUE_A1, DUE_D6, DUE_A0, and DUE_V_{in}.

Enclosure

The enclosure for the final version of the MPPT required a robust, waterproof, and clear box. The selected box that houses the entire circuit is not only waterproof but also abides by the International Electrotechnical Commission standards with an IP66 weatherproof rating, which states that the box is not only completely dust tight but it is protected against heavy waves and powerful jets of water. The base of the box is made from acrylonitrile butadiene styrene, and the top is clear and has a gasket lining to seal the box completely. According to this standard the type of box is used widely in electronics as a terminal box, supply switchboard, apparatus box, electrical power supply, and control box.

Further, this box has also been utilized on ship decks. When the three banana jacks for the connections to the ground, water pump, and photovoltaic panel were installed into the box, the Ingress Protection (IP) rating of the connection port is naturally reduced. A Lexan cover was also built over the connection port to protect this area from harsh weather conditions or possible water leaks.

There are wires soldered onto the banana plugs that run to the terminal blocks mounted on the printed circuit board with the exception of the solar panel connection. The solar panel connection to the banana plug, on the other hand, is made through a fuse for protection against a possible over current. The Arduino and Arduino ProtoShield package have been placed in the center of the box. There are four screws connecting a piece of Lexan to the enclosure to create a false bottom onto which the circuit boards have been

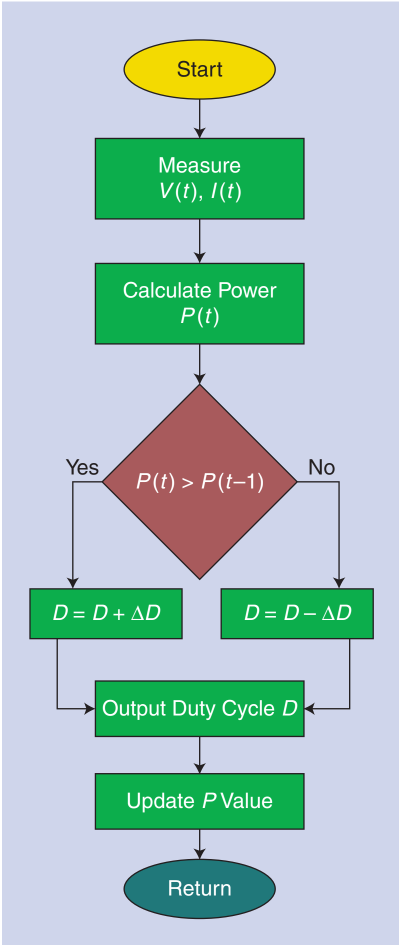


FIG6 A flowchart for the hill-climbing algorithm developed based on Yu and Lin's algorithm.

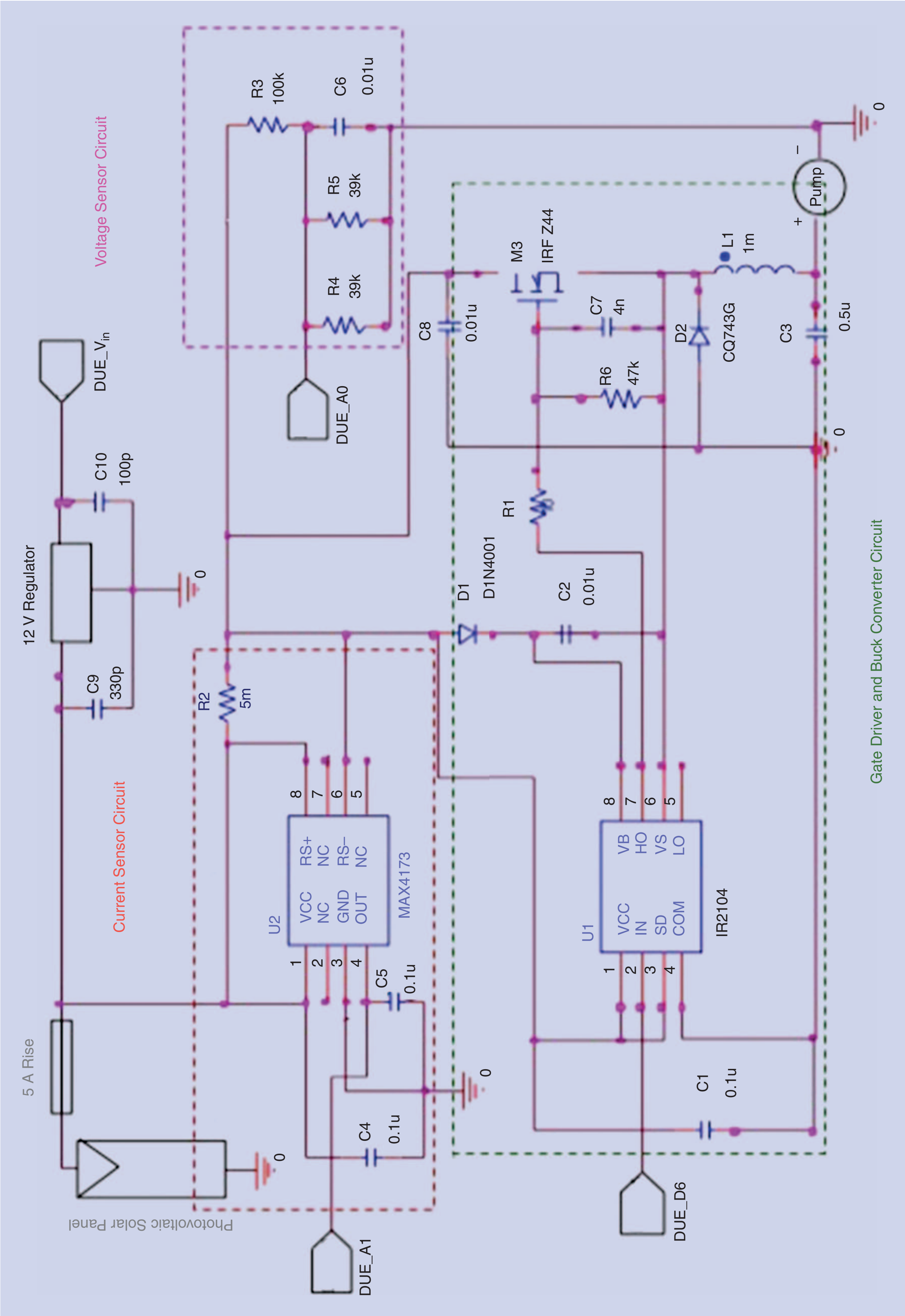


FIG7 The MPPT system schematic.

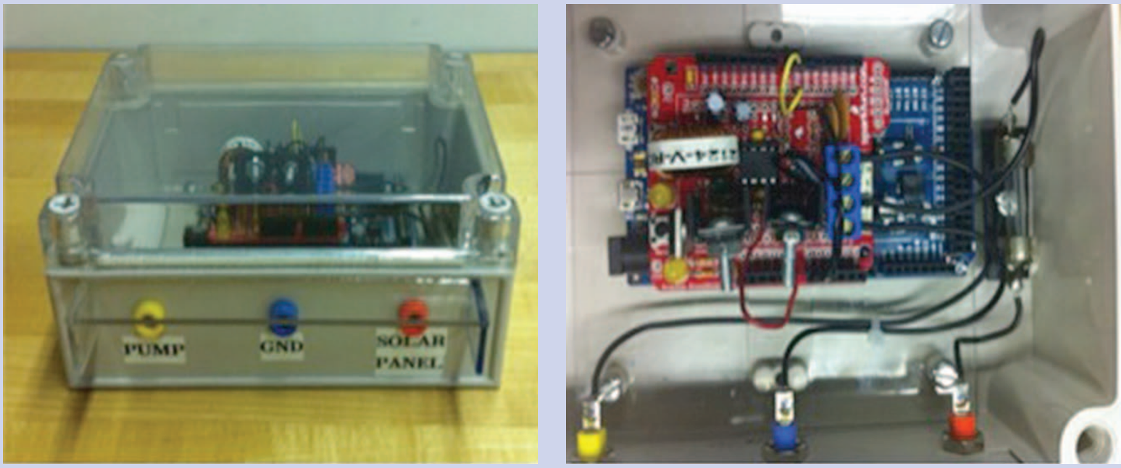


FIG8 The (a) side and (b) top views of the final package for the MPPT system.

mounted also using screws. The enclosure can be seen in Fig. 8.

Efficiency data of the final product

Prior to the photovoltaic performance validation, the MPPT system was tested for functionality with a high power supply. The functional test yielded that our system worked perfectly under constant current conditions as well as constant voltage conditions. During the functional test, the input and output voltage and current were collected and the efficiency was determined. The collected experimental data and calculated efficiencies are presented in Table 2. Moreover, Fig. 9 illustrates the efficiency plot of the MPPT system as a function of the output power.

The pump begins to work at approximately 1 A of output current, pumping water at a minimum rate. The efficiency observed at this instant is approximately 34.42%, which seems to be normal since the output power is low. As can be seen from Fig. 9, while the efficiency of 72.47% was achieved at 15.62 W, the maximum power of MPPT was at 21.55 W, as indicated in Table 2.

MPPT operations guide

First, the MPPT, the solar panel, and the water pump have to be connected, as shown in Fig. 10. The solar

TABLE 2. The MPPT system experimental data while testing with a power supply.

V_{IN} (V)	I_{IN} (A)	P_{IN} (W)	V_{OUT} (V)	I_{OUT} (A)	P_{OUT} (W)	EFFICIENCY (%)
10.06	0.353	3.55	1.219	1	1.21	34.32
11	0.439	4.82	1.829	1.09	1.99	41.28
12	0.639	7.66	3.328	1.3	4.33	56.42
13	0.721	9.373	4.29	1.44	6.18	65.91
14.03	0.893	12.53	5.656	1.76	9.95	79.45
15	1.248	18.72	7.18	1.88	13.5	72.12
15.99	1.348	21.55	8.01	1.95	15.62	72.47
17.17	1.249	21.44	7.63	1.94	14.80	69.02
18.02	1.21	21.80	7.89	1.98	15.62	71.65
18.88	0.955	18.03	7.01	1.91	13.39	74.26

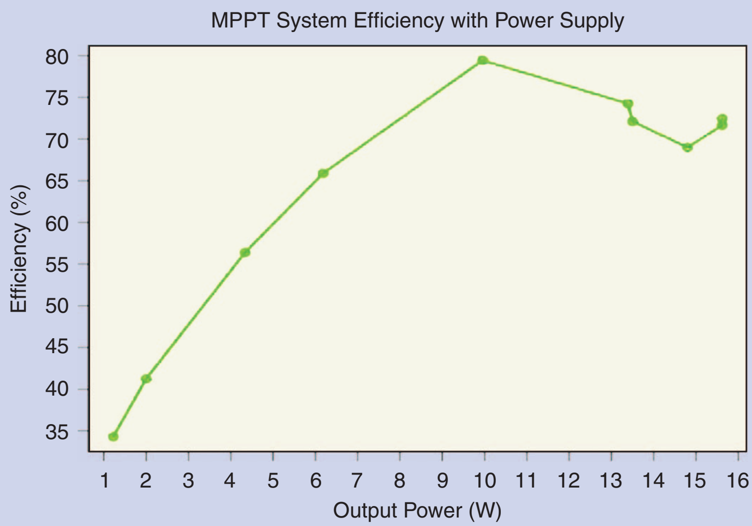


FIG9 The MPPT system efficiency plot as a function of the output power.

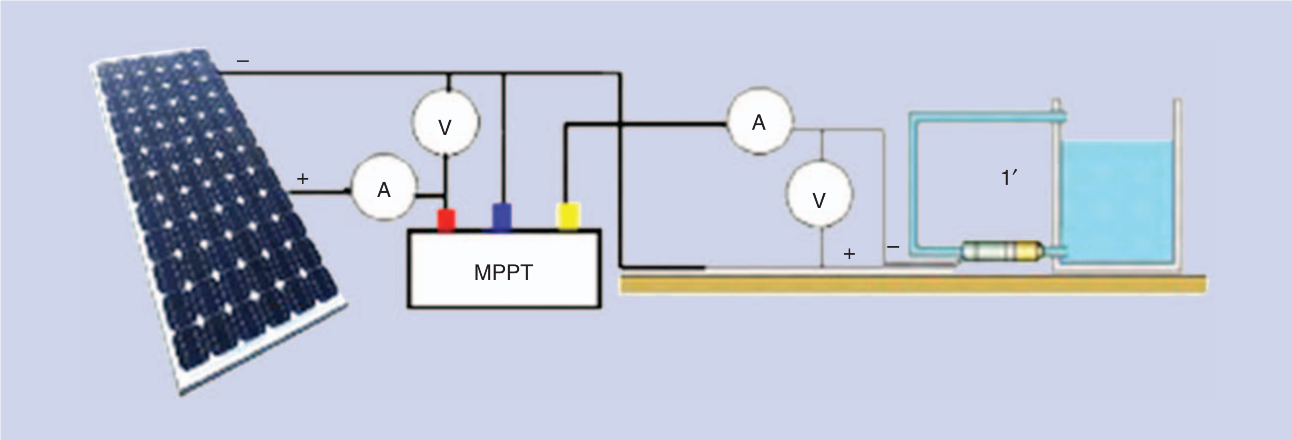


FIG10 The MPPT system connection.

panel "+" terminal is the red and the water pump "+" terminal is the yellow input connection. The generator and the load are connected to the same ground, thus, only one ground terminal has been placed in the case. Also, the voltmeters and ammeters were connected to the input and output of the converter circuit to make the voltage and current measurements for data collection purposes possible.

The Arduino board LEDs turn on at around 5 V, meaning that the Arduino has turned on, and the buck converter circuit begins functioning at 8-V input. The controlling components of the circuit receive power from the solar panel, so no further batteries or power supplies are required for the operation of the overall MPPT system. Anytime the input voltage of the Arduino is turned off, the program starts from the beginning with its duty cycle of 50%.

Conclusion

In the era of attempting to lead a green life, photovoltaic cells have become critical in producing electrical energy. Since a photovoltaic cell has a complex relationship among solar irradiance (W/m^2), temperature, and total resistance, it is the purpose of the MPPT to sample the output of the photovoltaic cell and apply a proper load resistance to adjust the solar operating voltage to one that is close to the maximum power point at changing atmospheric conditions. The MPPT system has become an essential component in evaluating the performance of photovoltaic power systems.

During this senior design project, we were able to successfully design, implement, and operate a functional MPPT system that was capable of producing a power efficiency of up to 79.45%. In the future, the MPPT could be implemented using a different software algorithm other than Yu and Lin's hill-climbing algorithm. The assessment of the newly developed algorithms could possibly increase the efficiency, thus producing a more effective MPPT system in the future.

Read more about it

- ARCO Solar Inc. "ARCO solar installation guide for solar electric modules," 1985.
- Arduino. (2013). Arduino Due Overview. [Online]. Available: http://arduino.cc/en/Main/ArduinoBoardDue
- T. Yu, and Y. Lin, (2010). A study on maximum power point tracking algorithms for photovoltaic systems. [Online]. Available: http://www.lhu.edu.tw/m/oaa/synthetic/publish/publish/30/4.%E4%BD%99%E5%AE%9A%E4%B8%ADA%20Study%20on%20Maximum%20Power%20Point%20Tracking%20Algorithms%20for%20Photovoltaic%20Systems.pdf
- ANSI/IEC. (2004). Iec degrees of protection provided by enclosures (ip code). [Online]. Available: http://www.nema.org/Standards/Complimentary-Documents/ANSI-IEC-60529.pdf

About the authors

A. Buke Hiziroglu (abhiziro@umich.edu) earned her B.S. degree in electrical engineering from Kettering University in 2014. At Kettering, she relaunched the IEEE Student Branch, served as vice president of Eta Kappa Nu, and was inducted into Tau Beta Pi. She is currently pursuing her M.S. degree in biomedical engineering at the University of Michigan, Ann Arbor.

Macy Payne (mpayne@mtu.edu) earned her B.S. degree in electrical engineering from Kettering University in 2013. While at Kettering, she served as treasurer of Eta Kappa Nu (Theta Epsilon) and was inducted into Sigma Pi Sigma. She is currently pursuing her M.S. degree in electrical engineering from Michigan Technological University.

Michael Glembotzki (m.glembo@yahoo.de) was a German exchange student at Kettering University from the Hochschule Esslingen in Germany. He previously completed a three-year skills training for the German power supply company EnBW in the field of mechatronics. Currently, he is a Bosch intern writing his undergraduate thesis on embedded webservers.

Mark Thompson (mthompso@kettering.edu) is a professor of electrical engineering at Kettering University. He teaches in the areas of electronic circuit and system design, including automotive electronic systems and alternative/renewable energy systems.