All-in-One Photovoltaic Sensor for Orlando Utilities Commission

Christian Avalos, Julian Gentry, Ryan Kehrmeyer, Zoran Kolega

Dept. of Electrical and Computer Engineering, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — As society moves forward, making a successful transition from fossil fuels to renewable sources of energy becomes increasingly important. One of the most popular methods of green power generation is solar, especially in the Sunshine State. In order to make the conversion to environmentally-friendly energy sources more seamless, it is necessary to have easy access to the data that is pertinent to solar panel performance. Our project aims to provide a consistent stream of accurate data regarding voltage, current, temperature, and irradiance to solar panel operators in order to aid them in solar array setup, adjustment, and maintenance.**

***Keywords* — Solar panel, database, sensor system, analog-to-digital conversion, collector node, DC power.**

I. Introduction

Solar farms are structured with strings of solar panels which share the same current and produce a shared voltage known as a string voltage. Similar to Christmas lights, this arrangement produces issues when one or more solar panels in a string become defective in producing the desired voltage output. The overall string voltage drop is not proportional to the amount of solar panels which are defective in a string. For example, if a string of 100 solar panels were to have one panel become defective, the string voltage drop could be much greater than 1% of the overall string voltage (these numbers are simply illustrative, and are not meant to be taken as real measurements). Panels can become defective when they are exposed to external factors, such as debris, shade, and water damage. Even electrical components, specifically wiring, can go awry and negatively impact the solar panel's performance. As such, we are motivated to design a device which can affordably and reliably measure voltage, current, and possibly temperature and irradiance to be able to quickly identify defective panels in a string.

A. Goals and Objectives

Our primary goal is to create several prototypes of a photovoltaic sensor usable in small- to medium-scale solar arrays that measure the voltage and amperage of the panels in order to determine panel health. The sensors will be uniformly distributed throughout a string of solar panels. This layout would allow a solar farm operator to determine the relative location of a faulty panel. As a result, on-site technicians would have a much easier time locating a defective panel, and could instead spend more time resolving the issue with the defective panel. Besides the benefits of easier repair, the sensor devices contribute in some other meaningful ways as well. The array's average performance should increase, providing more consistent power generation due to faster repairs or replacements. In addition, energy producers gain a more refined monitoring process and avoid surrendering fines to distribution partners when they cannot meet the agreed upon energy generation goals. Energy producers would also have access to more technical data regarding their solar panels, which would enable further insight as to the strengths and shortcomings of the manufacturer's product. Our secondary goal is to incorporate two supplementary measuring instruments to our original sensor design: a thermocouple capable of tracking internal panel temperature and a pyranometer that can detect incident light irradiance.

These additions separate our design from other iterations already in the market, and allow us to provide an extra layer of functionality and measurement to it in order to meet our sponsor's needs. Together, these devices will transmit data down the string to a local collector node that is hardwired to the producer's database. The producer can then monitor panel health and performance, optimizing the longevity and consistency of the array's generation.

B. Requirements and Specifications

There are certain requirements in performance and protection that must be met for the All-in-One Sensor to work. In regards to engineering, the sensor must be able to withstand 32V (DC) and 10A (DC), while being rated for up to 10W. In addition, the sensor must measure solar panel voltage and current with an accuracy of 5%. It must also transmit that data to the local database on the node within 10 seconds. It must have NEMA 4X rating [2] to withstand dust, wind, rain, and water splashing. The aforementioned specifications are summarized in Table 1.

TABLE I
ENGINEERING SPECIFICATIONS

Engineering Specifications	**Priority**
Must withstand 32V (DC)	High
Must be able to carry 10A (DC)	High
Must be rated for 10W	High
Must last at least one year of use	Moderate
Must be water resistant	Moderate
Must be under $20/sensor	Low
Demonstrable Specifications	**Deliverable**
Voltage measurement accuracy	Within 5% of real value
Current measurement accuracy	Within 5% of real value
Data transmission time	Accessible in local database within 10 seconds of measurement

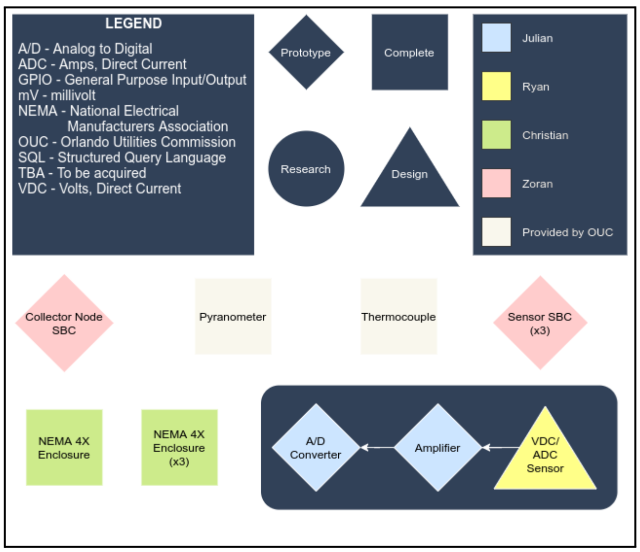


Fig. 1. Block diagram of the sensor system design.

II. PAST EFFORTS

A primary basis for our project is the TIDA-00640 design [1]. This reference design by Texas Instruments was built to provide an overview on how to construct a solar module monitoring and communication system. This design is where we got the inspiration for establishing the three major circuit elements: power supply unit, amplifier, and ADC. While the design application is similar to ours, we had to incorporate several new features, such as temperature measurement, irradiance measurement, and bluetooth connectivity.

III. HARDWARE DESIGN

Figure 1 shows the block diagram of our design, while Figure 2 shows a real-world visualization of the prototype when set up on solar panels.

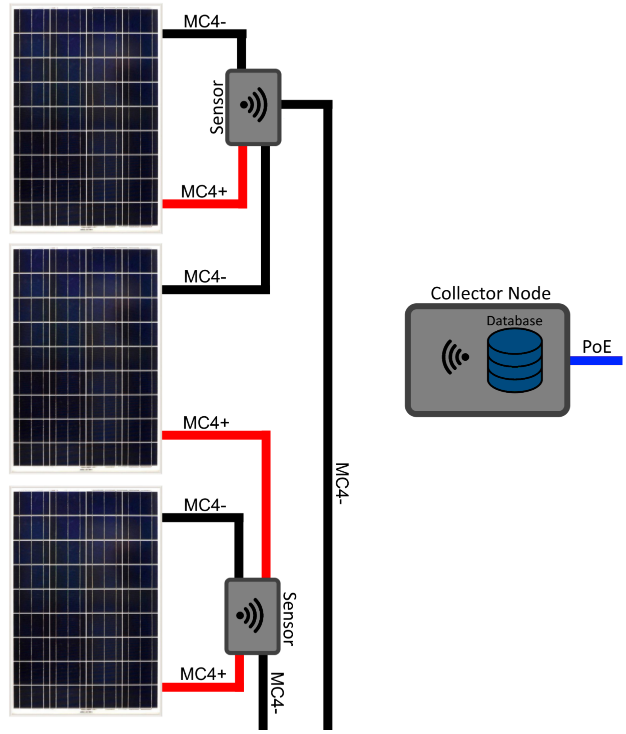


Fig. 2. Visualized prototype design. PoE supplies both power and an ethernet connection to the collector node. The database is stored locally on the collector node.

There are three main components of our printed circuit board, and two Raspberry Pi models for transmission and reception of the data. The first component is the power supply unit (PSU), the second is the amplifier (AMP), and the third is the analog-to-digital converter (ADC). The signals we measure are sent externally through the Raspberry Pi Zero W (RPi Zero) to the Raspberry Pi 4 Model B (node) where they are collected.

A. Power Supply Circuit

The power supply circuit is responsible for powering the active components in our design, such as the amplifier and the ADC. For our design, the power supply must have a large input range. The maximum output voltage from the solar panel would be about 32V (DC) and will fluctuate at different times of the day due to solar intermittence. Because of this, we decided to maintain overhead and establish the input voltage parameter (for the power supply) to be an upper limit of 40V (DC). Note that the precision of the output voltage of the power supply is not as crucial as the precision of its input voltage: this is because the devices involved in the circuit do not require a very specific voltage range with which to operate. Therefore, an output voltage of 3.3V was chosen, which is safely above the typical power supply lower-bound of 2.7V. Needless to say, the output voltage is expected to swing. However, the maximum supply voltage for the typical op-amp hardly drops below 5V, which preserves room for our power supply to adjust. Moreover, the ADC we selected shares the same upper and lower bound for its supply voltage. The final aspect needed in a PCB power supply design is the supply current. Typically, manufacturers do not divulge much information in their datasheets about supply current limits for op amps. Here, the same stood true about both op amps and the ADCs. Consequently, we decided to focus on avoiding hazardous heat dissipation amongst our electronics and chose a supply current of 0.1A (DC).

B. Amplifier Circuit

The amplifier circuit is responsible for amplifying the voltage and current values read from the solar panel. The voltage will be fed through a unity gain non-inverting amplifier, given that its magnitude is enough for the ADC to read. While the voltage syphoned from the panel need not be amplified, the current will be amplified by about 85x (non-inverted) in order for the ADC to register the value.

C. Analog-to-Digital Converter Circuit

The ADC circuit is the keystone to this project: it translates the analog voltage and current inputs to digital output, which is then communicated to the utility database.

IV. Software Design

The software design of the sensor system is split into three components which are responsible for collecting, converting, transmitting, and storing data collected by the hardware design.

A. Raspberry Pi Zero W Python Script

The Python script which runs on the RPi Zero in each sensor receives the digital values which have been converted from analog by the ADC circuit, translates those values to measured units, and transmits the data to the node upon receiving a request to do so. Figure 3 shows a flowchart of this Python script.

Before a sensor begins collecting data, it establishes a Bluetooth connection with the node to enable wireless transmission. It then awaits a signal from the node. Once a request is received, the script obtains the values on the four channels of the ADC which represent the measured current, voltage, temperature, and irradiance.

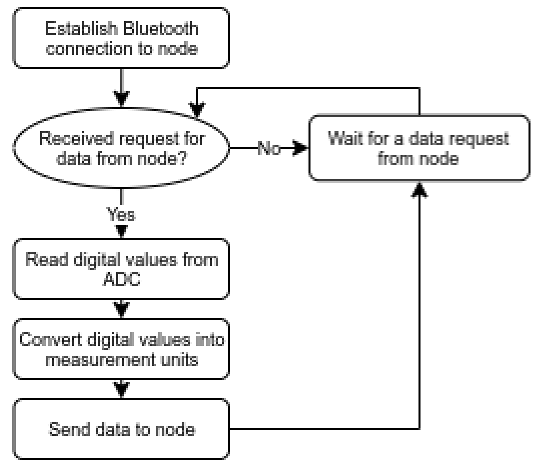


Fig. 3. Flowchart of the Python script which runs on the Raspberry Pi Zero W in the sensor.

With the known reference voltage of 5V and 10-bit resolution of the MCP3008 ADC, we know that each of the 1024 steps represents approximately 4.88mV. With this in mind, the script multiplies each digital value by (5.0 / 1024) to convert the digital values to the input voltage received by the ADC. The process for converting each of these voltage values to the real measured unit for each measurement varies.

The voltage value is divided by the gain factor of the amplifier circuit to be converted to the measured value. To get the measured current value, the voltage value for the current is also divided by the amplifier circuit gain as well as the known shunt resistor value of 1 mΩ. The temperature and irradiance measurements rely on the manufacturer's specifications of each device to convert their values to the correct units. As an example, the pyranometer used in our design specifies that each millivolt represents $5W/m^2$ of incident light irradiance. As such, the irradiance voltage value is multiplied by 5000 to obtain the measured incident light irradiance value. As mentioned, the temperature value follows the same process, except it uses the conversion factor specified by the thermocouple in our design.

After all the digital data is converted to measured units, the script transmits the data to the node. The sensor then waits for another request for data.

B. Node Python Script

The Python script which runs on the collector node is the collection point for the data transmitted from the sensors. It generates timestamps for the data, keeps the program in proper time, and stores the data in the database. Figure 4 shows a flowchart of this Python script.

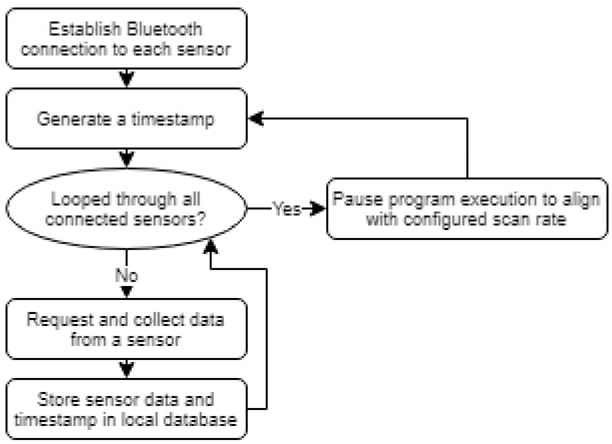


Fig. 4. Flowchart of the Python script which runs on the Raspberry Pi 4 Model B node.

The node's script begins by establishing a Bluetooth connection with each sensor assigned to it, then the data collection loop begins. First, a timestamp is generated. Next, an inner loop begins to request data from each sensor. When data is received from a sensor, the timestamp is added into the received JavaScript object notation (JSON) object, and inserted into the database. Finally, the script halts its execution for the remaining time before the next iteration of data collection, as configured by the scan rate which is designed to be as fast as two seconds.

C. Database

The database is responsible for storing the data collected in an easily accessible and transferable format. A local instance of MongoDB on the node accomplishes this task. It consists of a single collection, in which each instance of data collected from each sensor at each time interval is its own document. The JSON object in each document contains six fields: voltage, current, temperature, irradiance, timestamp, and sensor_id. The sensor_id field contains the Bluetooth MAC address which is unique to the sensor that the data was collected from, enabling the data to be traced back to a panel or set of panels.

V. Hardware Selection

With our required specifications in mind, we were able to form criteria that we used in selecting our hardware. To reiterate, we needed to be able to measure the panel voltage and current within a 5% tolerance and successfully transmit the data to the local database within 10 seconds.

A. Power Supply Circuit

Designing a power source in our circuit board is crucial due to the variety of active elements that must be powered to operate. When designing the power source in our circuit board, we prioritized maintaining a level of efficiency and high performance. However, our highest priority was making sure that our onboard power supply would have a wide input voltage range, considering the 32V (DC) maximum voltage we expect from the solar panel, as well as the intermittence of solar power. Instead of designing each component from scratch, we decided to utilize Texas Instruments' (TI) WEBENCH tool, which offers a variety of recommendations of schematics based upon our detailed parameters.

In Table 2, we have summarized the five primary designs that we considered for our onboard power supply. Ultimately, we decided to implement the LMR50410X based upon its high efficiency and low BOM cost (highlighted in green).

Table II
Power Supply Review

Part #	**Efficiency (%)**	**BOM Cost ($)**	**Footprint (mm^2)**
LMR36506RF3	64	1.29	112
LMR50410X	81.6	0.70	142
LMZM23600V3	79.5	2.32	41
TPSM265R1V3	71.7	1.98	40
LM317	9.8	0.4	357

B. Amplifier Circuit

In our design, we established that we would require two types of amplification: one requiring a unity gain and another requiring resistor ratios. Therefore, we aimed to implement a dual package amplifier in our design.

In Table 3, we have summarized the five primary designs that we considered for our amplifier circuit. While all of the op amps met our specifications, they all had their respective shortcomings. The NE5532 and LM741 were much more costly, which did not meet our cost expectations. The LM358B was the most robust, yet overqualified for our application. Lastly, the RC455B was very similar to the TL072, yet the latter is a more widely used op amp and therefore is in greater supply and has more actual designs to use as reference.

TABLE III
AMPLIFIER REVIEW

Op Amp	Channels	Max Supply Voltage (V)	Output Current (mA)	Cost ($)
TL072	2	30	10	0.075
NE5532	2	30	38	0.296
RC4558	2	30	10	0.055
LM358B	2	36	30	0.051
LM741	1	44	25	0.236

C. Analog-to-Digital Converter Circuit

To reiterate, the ADC circuit is responsible for communicating the analog voltage and current values measured from the panel to the local database. Because we have decided to employ the Raspberry Pi Zero W single-board computer (SBC), we explored three different ADCs that are seamlessly compatible with Raspberry Pi devices. This search is summarized in Table 4.

After considering all three ADCs, the most practical solution for our system is the MCP3008. It has the simplest design, which can be easily incorporated in our system. Although the ADSx series ADCs both maintain higher resolutions, it is unnecessary for our scope and not worth the significant increase in price. Moreover, the MCP3008 implements an SPI interface, whereas the ADSx ADCs utilize an I2C interface. For our project, SPI's ability to communicate with its peripherals quickly and effectively (via its full-duplex configuration) qualifies the MCP3008 as the preferred ADC for our project.

TABLE IV
ADC REVIEW

ADC Device	MCP3008	ADS1015	ADS1115
Channels	8	4	4
Resolution	10-bit	12-bit	16-bit
Sample Rate	200,000	3,300	860
Cost ($)	3.75	9.95	14.95
Interface	SPI	I2C	I2C

D. Sensor SBC

Each sensor in the design is responsible for collecting at least two and up to four data values. After the hardware design converts the string voltage and current to values which can be input to the ADC, the sensor SBC is responsible for collecting the digital output data. In addition to collecting digital data, the sensor SBC must also be capable of wirelessly transmitting the data to the collector node via Bluetooth. Table 5 shows a comparison of the choices of SBCs which were considered to be used in the sensor, with our choice highlighted in green.

TABLE V
COMPARISON OF CONSIDERED SENSOR SBCS

Properties	Raspberry Pi Zero W	BPI-M2 Zero	Orange Pi i96
Bluetooth Capability	Bluetooth 4.1, BLE	Bluetooth 4.0	Bluetooth 2.1
Operating Temp. Range	0 ºC - 70 ºC	N/A	-10 ºC - 65 ºC
Price	$10.00	$24.99	$9.99

E. Node SBC

The collector node acts as a nexus for the data collected by the sensors in a string of solar panels. Its hardwired ethernet connection also makes it a reliable access point for the collected data. The database is stored on the node, of which the sole component is the SBC. Table 6 shows a comparison of the SBCs which were considered to be used as the collector node.

TABLE VI
COMPARISON OF CONSIDERED NODE SBCS

Characteristic	Raspberry Pi 4 Model B	BPI-M5	NanoPi M4B
PoE Support	Yes - w/ PoE HAT ($20.00)	No	No
Bluetooth Capability	Bluetooth 5.0, BLE	None	Bluetooth 5.0
Operating Temp. Range	0ºC - 70ºC	N/A	-20ºC - 70ºC
Price	$30.00	$69.99	$70.00

F. Enclosure

Once our printed circuit board has been tested and proven, we will need to install it within its own enclosure. This is because the sensor will be implemented outdoors, where a number of harsh conditions are expected: harsh weather, extreme temperatures, rain, and exposure to dust and dirt. Note that exposure to these conditions contribute to the corrosion and overall degradation of electronic devices. To prevent this, we will utilize an enclosure that satisfies the aforementioned NEMA standards. As previously stated, the NEMA 4X rated enclosures provide the most appropriate support for our sponsor's needs. The NEMA 4X rating certifies that the enclosure is designed to provide protection against dust, water, and ice, along with an extent of insulation [2].

Our enclosure must also provide more than enough room for our printed circuit board. Although our printed circuit board is about 3" by 3", and the RPi Zero is about 2" by 1", we need ample room remaining in the enclosure for a variety of reasons. For starters, there needs to be enough room in the enclosure for installation. If the enclosure only provides an inch of room between the circuit board and the enclosure wall, then the installation will prove to be much more difficult. Moreover, the same logic applies to any maintenance the board may require. Another reason for leaving extra spacing in the enclosure is for ventilation purposes. Because of the relatively high levels of voltage and current pumping through each circuit board, there should be, for lack of a better word, breathing room for the circuit board to prevent overheating.

After reviewing all of the options summarized in Table 7, our team decided to use the versa-mount polycarbonate washdown enclosure with the opaque panel-door. The polycarbonate corrosion-resistant washdown enclosure was a minimalist design, which was attractive. However, it's higher cost did not appropriately account for its lack of design features, compared to the second option. The versa-mount polycarbonate washdown enclosure is also designed to handle many more outdoor factors via its four different ratings. The size of the second option was also much more appropriate for our electronic devices. Considering that our two electronic devices combined fail to exceed 5" in length, it is not necessary to select an enclosure that is more than twice that size. As for the decision between an opaque door or translucent door, we decided that incorporating the see-through door would be a superfluous addition with an unnecessary increase in cost.

TABLE VII
ENCLOSURE REVIEW

Polycarbonate Corrosion-Resistant Washdown Enclosure	**Rating**	**Starting Size (in)**	**Cost ($)**
Normal	NEMA 4X NEMA 13	11" x 8" x 7"	$111.08
Versa-Mount	IP66 NEMA 3S NEMA 4X NEMA 13	6.5" x 6.5" x 5.5"	$71.97
See-Through Cover	NEMA 4X NEMA 13	11" x 8" x 7"	$111.08
Versa-Mount and See-Through Cover	IP66 NEMA 3S NEMA 4X NEMA 13	6.5" x 6.5" x 5.5"	$77.07

VI. SOFTWARE SELECTION

When deciding between several choices in the software design user friendliness was the first consideration. Countless software options were considered in arriving at the final software design, of which the most impactful will be discussed here.

A. Operating Systems

Both the RPi Zero and the RPi 4B node require an operating system. Raspberry Pi OS is the obvious choice for a user friendly operating system for both devices. It is extraordinarily lightweight to maximize the performance of the devices, and features a simple graphical user interface (GUI). Unfortunately for our design it is only available in a 32-bit distribution, which makes it incompatible with larger MongoDB databases. As such, the RPi Zeros in the sensors run Raspberry Pi OS, while the node runs Ubuntu Desktop 64-bit [3].

B. Higher Level Software

Among the higher level software choices in the software design is the database. MongoDB was used for its simple NoSQL create, read, update, and delete operations. It also has a straightforward hierarchical structure which starts with databases at the top, collections, and then documents which are made up of JSON objects, which are lists of key-value pairs. Sending queries to the database uses the same JSON format, making it simple to find specific values or sort by timestamps. The complete package makes for an accessible database design for solar farm operators to be able to further integrate into their existing networks.

Python was used to write the scripts which perform the data collection, transmission, and storage. It is the language of choice for Raspberry Pi devices and the like, and it has vast community support to boot. To enable database compatibility with MongoDB, the PyMongo library is used.

As Bluetooth is core to our software design, having a robust interface for it was critical. Though Python 3.3 and newer versions support Bluetooth via sockets, it lacks some features like device discovery and service advertisements. To enable our sensors to more seamlessly connect with the node, the PyBluez Bluetooth Python extension module is used.

VII. TESTING

Testing began with breadboarding the major circuit elements: the amplifier circuit, the power supply circuit, and the ADC circuit. All three circuits were tested with a simulated photovoltaic input with a DC power supply. After successful tests with the DC power supply, the design was tested with a 40W power supply to more closely match the expected power of the solar array. Unfortunately, the 40W power supply far exceeded the breadboard's capabilities and subsequently fried it and the connected components. This led to a stronger medium for testing: protoboards.

Reimplementing the same circuits on the protoboards revealed some unexpected issues, which were likely due to reuse of components which had been damaged during the breadboard testing. Considering our prior success and lack of time, we decided to forego further protoboard testing and ordered printed circuit boards (PCBs). With the PCBs ordered, we also had to reorder the other circuit components. Unfortunately, the LMR50410X became backordered, causing us to order its sister components: LMR14010ADDCT and LV2862XLVDDCR. These two parts matched the original's specifications quite well. As a result, we did have to quickly revise our original PCB design and order a new batch (PCB V.2).

With the PCBs assembled, and using the DC power supply to test them, all of the major circuit elements were functioning properly, with the exception of the current side of the amplifier. This issue existed on both versions of the PCB, leading us to believe that the DC power supply couldn't output enough current for the amplifier to recognize. Field testing the current set of PCBs by plugging them into the solar array was the next step.

When implementing the boards with the solar array, a strange problem occurred: the board was only receiving about 1.6V instead of the expected 34V. With the help of our sponsor, we began troubleshooting the issue. We confirmed that the panel was outputting 34V and that the DC optimizer was outputting near 0V when unplugged, as expected. We tried plugging into a different panel on the string, but to no avail. With our options limited, we decided to try plugging a board directly to the panel, expecting to receive all 34V to the board without the regulation of the DC optimizer. Fortunately, our board was able to receive all 34V with no damage. However, this revealed that the issue lay somewhere between the interaction of our board and the DC optimizer, which left our team and our sponsor puzzled.

VIII. Results

A. Voltage Measurement Accuracy

The final prototype contains a fully functional power supply unit (PSU). As we fed the PSU different input voltages, the PSU consistently produced a steady 5V output, seen in Table 8.

Table VIII
Voltage Output Accuracy

Voltage (V)	8.000	16.000	24.000	32.000
PSU Output (V)	5.000	5.000	5.000	5.000
Percent Error	0%	0%	0%	0%

With a steady 5V power supply from the PSU, the amplifier functions properly as well. With an input of 8V, the voltage divider steps this down to 1V, and the amplifier amplifies the 1V with a unity gain, seen in Figure 5.

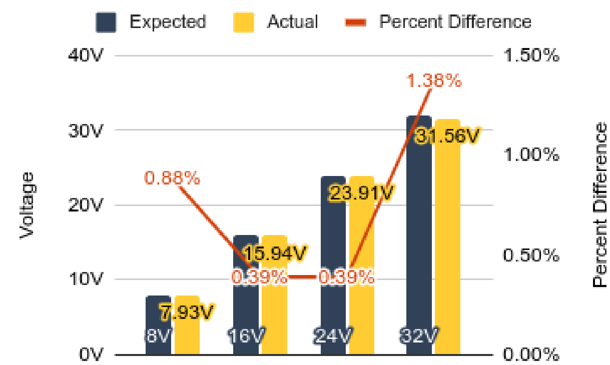


Fig. 5. Expected vs. actual voltage recorded in the database.

It is crucial to note the integral bypass capacitor connected to the amplifier's V_{DD}. Without this capacitor, the 5V supply would experience ripples, which could increase the supply voltage above the amplifier's rated input. This could cause a short in the amplifier.

B. Current Measurement Accuracy

The amplifier is responsible for measuring the current flowing through the panel. However, our amplifier could not get an accurate reading of the input current. This inaccuracy could be the result of a few possibilities. First, when using the DC power supply, we can see how much current is being drawn by the sensor. When simulating the PV input with the DC power supply, the sensor would only be drawing about 1mA. This low current consumption could be so small that the amplifier circuit could not register the value properly. Another possibility could be the input offset voltage of the amplifier. The amplifier we are using (TL072) may have an input offset voltage that is too high, causing the amplifier to malfunction.

C. Data Transmission Time

Although we were able to obtain a functioning ADC circuit, we were not able to transmit the temperature values. Even while using the thermocouple provided by OUC, the readings from the thermocouple were always inaccurate. In fact, the voltage measured between the two leads was inaccurate, which is likely evidence that the product was defective. However, this was a low priority for our sensor and could be easily resolved by obtaining a different thermocouple.

The data transmission time is measured from when the node requests data from a sensor, to when that data is inserted into the local database on the node. Tracking this transmission time across 10,000 data transfers revealed an average of 50 milliseconds. The maximum transfer time was 672 milliseconds, though this only occurred on the first data transfer. All other transfer times were very near the average, leaving the transfer times within the 10 second specification.

IX. Conclusion

Using the TIDA-00640 reference design [1], we created a sensor that was able to measure the irradiance and the voltage across a panel with 5% accuracy and communicate the readings to the utility database. With minor modifications, the ambient temperature of the panel could also be communicated to the database.

To reiterate, the current-measurement issue is something that can continue to be explored in the next iteration of the project. The team learned important lessons on always being aware of the power ratings of the relevant components and handling power electronics. Once the current-measurement is perfected, the sensor will be ready to be installed along solar panel strings and assist utility-based companies and other solar farms with optimizing their solar power output.

Acknowledgement

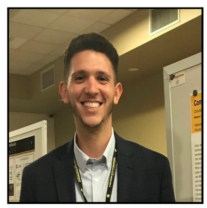
The authors wish to acknowledge and thank the assistance and support of Dr. Mark Steiner from the University of Central Florida. We also extend our thanks and gratitude to Rubin York and the members of the Orlando Utilities Commission in Orlando, FL for funding this project. We offer our thanks and appreciation to the professors of the evaluation committee who have agreed to review and critique our project.

Biographies

Julian Gentry will be graduating with the degree of Bachelor of Science in Electrical Engineering with University Honors in August 2021. His work experience includes Lockheed Martin CWEP Program and CRB Consulting Engineers. His plans are to transition to a full-time position with CRB, move to North Carolina, start a family, and begin work as a power engineer.

Ryan Kehrmeyer is a graduating Electrical Engineering student from the University of Central Florida. His main interests lie in the fields of renewable power generation, automation and semiconductor materials. He is currently seeking full-time employment in the Orlando and Tampa areas.

Zoran Kolega will be graduating with the degree of Bachelor of Science in Computer Engineering with University Honors in August 2021. He is currently researching the impact of persistent memory on computer applications. He has been offered and accepted admission to the Computer Science MS program at UCF for the Fall 2021 semester.



Christian Avalos will be graduating with a Bachelor's degree in Electrical Engineering with a focus on Power Systems & Renewable Energy in August 2021. During his undergrad, Christian conducted solar cell research and interned with Mitsubishi Power. In the fall, Christian will attend Arizona State University in pursuit of his Master's degree, where he plans on continuing his photovoltaic research and hopes to find work with a solar manufacturing company.

References

[1] Texas Instruments, "Voltage, Current, and Temperature Monitoring for Solar Module Level Power Electronics," TIDUCM3 datasheet, Dec. 2016. [Online]. Available: https://www.ti.com/lit/ug/tiducm3/tiducm3.pdf?ts=1627178369589&ref_url=https%253A%252F%252Fwww.ti.com%252Ftool%252FTIDA-00640 . [Accessed: 02-Mar-2021].

[2] "NEMA Ratings For Enclosures," NEMA Enclosures. [Online]. Available: https://www.nemaenclosures.com/enclosure-ratings/nema-rated-enclosures.html. [Accessed: 21-Mar-2021]

[3] Shawn, "Raspberry Pi Operating Systems (OS) - Which one to use in 2020?," Seeed Studio, 2020. [Online]. Available: https://www.seeedstudio.com/blog/2019/10/29/raspberry-pi-operating-systems-os-which-one-to-use/. [Accessed: 08-Apr-2021].