Paper ID #35068

Prototype Development for Adaptive Solar Tracking and Optimization of Data Communication Protocol

Mr. James D. Kaul, Marshall University CCSE

Graduated from Woodrow Wilson High School & Raleigh County Academy of Careers & Technology in 2003 with a concentration in Computer Networking and Electronics. Graduated with Honors from Mountwest Community & Technical College in Applied Engineering Design Technology in Spring 2014. Joined Marshall University Fall 2014, and graduated with a Bachelors of Electrical/Computer Engineering in Dec. 2020.

Mr. Gregory Dovel Weed
Jared Cunningham
Imtiaz Ahmed
Dr. Wook-Sung Yoo, Marshall University

Dr. Wook-Sung Yoo is chair and professor in the Department of Computer Sciences and Electrical Engineering at Marshall University. He received his Ph.D. from the Florida Institute of Technology in 1995. He has a diverse academic background in Computer Science, Electrical and Computer Engineering, Information Systems, Cybersecurity, Data Science, Software Engineering, Health Informatics, Bioinformatics, Dentistry, Dental Informatics programs at various public and private universities with more than 15 years of administrative experience as a chair/program director and 25 years of teaching, research, service, and industrial experiences.

©American Society for Engineering Education, 2021

Adaptive Solar Energy Harvesting and Data Transmission

James Kaul, Greg Weed, Jared Cunningham, Alisha Sue Pettit, Imtiaz Ahmed, Wook-Sung Yoo
Computer Sciences and Electrical Electrical Enginering
Marshall University, Huntington, WV, USA

Abstract

A prototype for an adaptive solar tracking and efficient data communication system empowered by the harvested solar energy was developed by a capstone project team at Marshall University. The prototype is developed on Raspberry Pi and Arduino development boards and the overall system comprises a solar tracking module, power banks, data transmitter, receiver supported by omni-directional antennas, and web-portal at the receiver. The harvested energy powers up an integrated sensor-mounted data collection and wireless transmitter module. The algorithm for the optimal control of solar tracking was developed in Python using open source DNN modules such as TensorFlow (Lite) to implement LSTM-assisted DNN for optimization of data transmission instances. A web-portal with a database shows the real-time data collection at the receiver end and exhibits dynamic performance analysis based on the collected information. The experimental results show the prototype significantly enhanced the efficiency of energy harvesting. Future enhancement is discussed.

Keywords

Energy Harvesting, Adaptive Solar Tracking, Data Communication, Machine Learning

Introduction

Lithium-Polymer (LiPo) batteries have revolutionized electrical storage technology. They are lighter than traditional alkaline batteries, provide more charge/discharge cycles, last longer before failure, and among several other benefits they do not carry the same corrosion and environmental concerns as alkaline batteries do.[1] However, despite that fact that LiPo prices have fallen significantly over the past 10 years these batteries still come at a price premium over their traditional counterparts. To further complicate the matter, there is very little capacity for the recycling or handling of LiPo batteries, with only 5% of used batteries being recycled in 2020.[2] With the added cost and the impact on the environment, it is desirable to design mobile electronics in a way that will maximize the lifespan of the battery.

LiPo batteries have been studied for their longevity by the NERL[3], and two of the major contributors to LiPo battery failure are the operating temperature and the percent discharge during a cycle. Temperature had the greatest impact on capacity of the battery over time, with warmer average temperatures effecting the capacity by 54% on average. Meanwhile, the percent discharge between cycles had the greatest impact on the longevity of the battery, seeing a significant reduction in life between 50% and 80% discharge per cycle. In an applied electronics power system these two factors work together against the lifespan of the battery, as the

© American Society for Engineering Education, 2021

maximum capacity degrades over time, the electrical load will discharge the battery further and further until the battery can no longer hold sufficient charge and must be replaced.

Controlling these two variables is the key to maximizing battery lifespan. In this work, an environmental data collection device was developed to be reliant upon renewable solar energy, use that energy in the most efficient way possible, and require minimal human interaction. Radio frequency (RF) communications were employed in this design to transmit collected data to an access point at a discrete time, creating an event that greatly increases the electrical demand for a short period that can be provided by either the solar panel or the LiPo battery. In order to reduce the load and heat applied to the battery, it is desirable to only transmit during times when the solar panel array can provide 100% of the necessary power for the transmission. In general, data transmission at pre-defined instances while being powered by the solar energy cannot make the efficient utilizations of the harvested energy.

On the other hand, in this project, we deployed solar tracking system and calculated optimal instances of data transmission that is designed by deep learning algorithm.[4] We exploited long short-term memory (LSTM) assisted deep neural network, which is trained over a time span and predicted the optimal data transmission instance in a day. [5]

The weather is notoriously hard to predict; however, some patterns do appear over time. If such a pattern can be established, it is possible to predict with some success when optimal conditions will exist for the solar panels. This work uses the advantage of collected data to establish this kind of pattern with the use of a machine learning algorithm.

Our Approach

In this research, a system was designed and developed that includes the integration of a sun tracking photovoltaic/LiPo battery power system, constant and variable electrical loads, data collection, embedded system on chip computing with data handling software, and a deep learning algorithm using TensorFlow[6] to predict optimal conditions over time to perform tasks that require higher than normal power consumption.

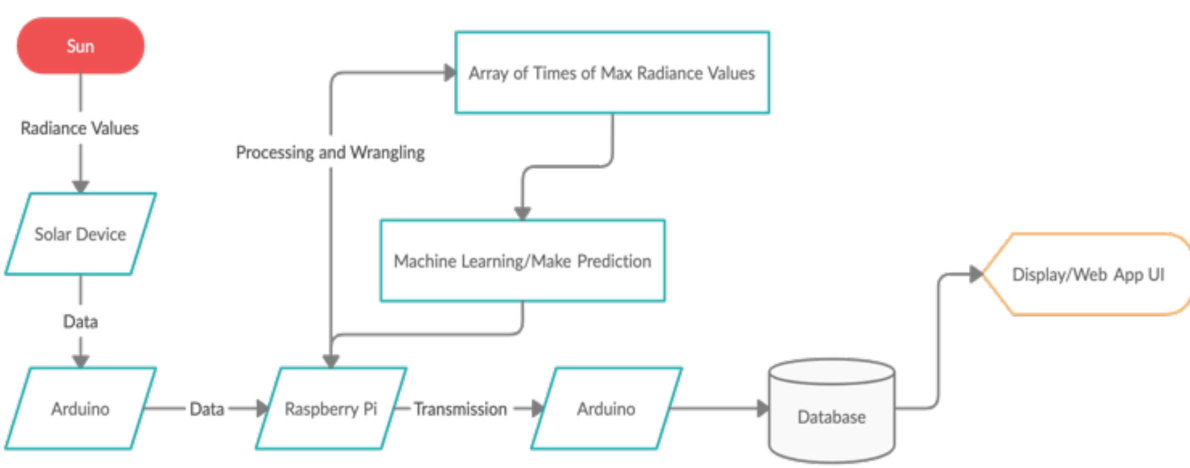

Fig. 1: Flow of Data from Start to Finish

© American Society for Engineering Education, 2021

Data Collection

An Arduino microprocessor connected to three sensors: radiance, temperature and humidity, and a real-time clock (RTC) module are used to gather data from the environment. Additionally, a Brown Dog Gadgets Dual Axis Solar Tracker (or similar dual axis sun tracking system) will provide feedback control for orienting the solar panels at a normal angle to the sun. This data is sent from the Arduino to a Raspberry Pi (RPi) using utf-8 encoding via a serial connection for processing and retention.

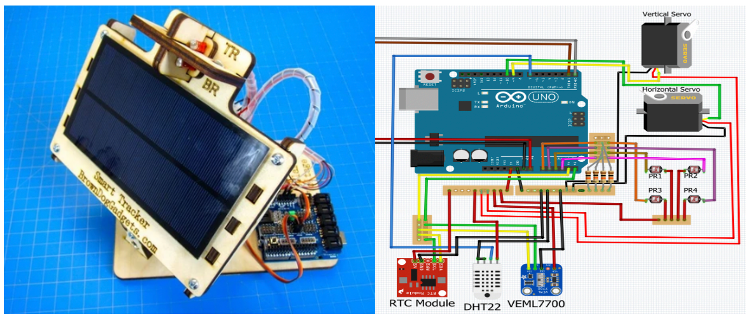

Fig. 2: Data Collection Subsystem

Arduino boards are single thread boards that can only run one code at a time. Because of this limitation, and the requirement to run the sensors at a different sample rate than the solar tracking, a time delay using the millis() command allows the Solar Tracking and the Sensor Reading/Sending functions to run concurrently at different rates.[7] Solar tracking is performed with each loop of the controller, where the outputs of the 4 photoresistors are compared and if the difference is over a certain threshold, the appropriate axial servo is pulsed to rotate the panel. Although the sun moves at a rate of 0.25° per minute across the sky, the design is not precise enough to allow such small changes, and the shadow cast by the sensor divider must be sufficiently large enough to cover one of the sensors. If the sensors are all receiving equal amounts of sunlight, no outputs are issued, and an insignificant amount of power is used between pulses. Sensor reading/sending occurs every sixteen minutes in order to limit the memory usage on the RPi. This task is conducted by comparing the current millis() returned value to the last recorded millis() value. If the difference is more than 960,000 milliseconds, the sensors are read and sent to the RPi. A more frequent sampling rate will allow more accurate predictions from the machine/deep learning algorithm, but the processing time will increase dramatically.

Processing Data

When the utf-8 data is received by the Raspberry Pi from the Arduino, the information is exported into a text file for storage. All the readings are also read into a data frame by the *processData* function. In this function, an object holding all of the attributes of the readings is created and put into a list. These readings are then compared based on their radiance values to determine the maximum radiance value for each day. This comparison produces a list of times that correspond to the maximum radiance value for each day of data that comes through the function. For instance, the output list is generated in the form of [1130, 1230, 1350, 1410], using a 4-digit number without colons for the time of maximum radiance for that day.

© American Society for Engineering Education, 2021

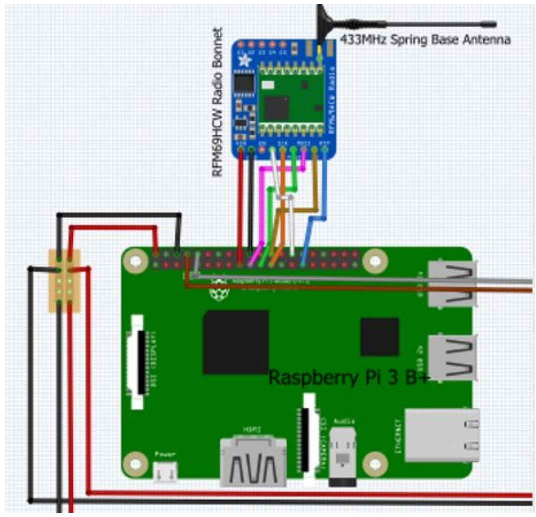

Fig. 3: RPi used for Data Processing, Prediction Algorithm, and Transmission

Prediction Algorithm

Two weeks of initial data was gathered with the use of a LUX sensor with a timestamp and used to train an LSTM deep neural networkneural network using TensorFlow libraries. During operation, the algorithm passes the past recorded maximum LUX values through the neural network to predict a time within a 3 day window that offers the best chance of maximum radiance. Once this prediction is generated, a crontab triggers the transmission and is updated with the new time, and a prediction log is created and written to a text file.

Wireless Transmission

Transmission automatically takes place at the time and date specified by the cron job. A wireless RF scheme for a Hopewell RFM69HCW[8] at 433MHz, using FSK modulation was developed for this project because the lower frequency requires less power to transmit, travels further without signal degradation, and does not require complex handshaking or connection protocols like those found with TCP/IP or Bluetooth. The software libraries are developed so that node to node communication requiring encryption and acknowledgements is optional, and the system can be set up as a simple broadcast system for further power savings.

The buffer on the radio chip has a 64 byte buffer for packets, which includes a 4 byte header and 60 bytes of transmittable data. The text files that hold the prediction logs and the data are read through and divided into 60 byte segments. These packets are then transmitted at 20 dBm (100mW) with a 100ms delay between packets to not overload the buffer at the receiver.

At the receiver, another RFM69HCW with an Arduino microcontroller to be connected to the USB port on a suitable workstation. This microcontroller simply listens for the packets and prints them in utf-8 encoding along the serial USB connection. The workstation runs a python script that handles each of the packets and writes them to either the prediction log text file, or the data text file, based on Boolean values that are triggered by text phrases sent by the transmitter.

© American Society for Engineering Education, 2021

Database & HMI

The text files appended by the receiver script is the long-term data storage that can be used by the end user for analysis. Using PHP and MySQL, a web browser based Human Machine Interface (HMI) has been developed to format and display the information in the text files. Navigation through the HMI will display prediction logs, the sensor data at given dates and times, as well as the maximum radiance for each day. Accuracy of the predictions can be validated by comparing the predicted times with the actual readings. Although this project focused on a local machine, this HMI could be deployed as a website if the database were hosted on a sever, providing additional remote use to the end user.

Experimental Results

The goal of this project is to mitigate power demand from a LiPo battery to a photovoltaic source during a high demand event. To accomplish this, a deep learning algorithm is used to predict the best time to trigger this process, so that the available LUX is at its maximum value for a given day. With no long-term testing analysis possible, a model using collected data is used to simulate a dynamically changing event time against a static event time. The LUX values used to train the neural network has been compiled on a spreadsheet, columns by day and rows by time. The LUX value has been converted into available watts per square meter and an efficiency of 20% was assumed for the solar panels. Assuming a 10-minute window, and a static event time of 1:30PM each day, a dynamic time that will represent an accurate prediction made by the machine learning algorithm shows the average available power to be 166% higher than the static time. More importantly, over the course of these 15 days, 6 of those days shift the energy supplied partially or fully from battery power, to 100% solar energy, while 5 of those days would not have triggered the transmission at all. These observations are displayed in Fig. 4.

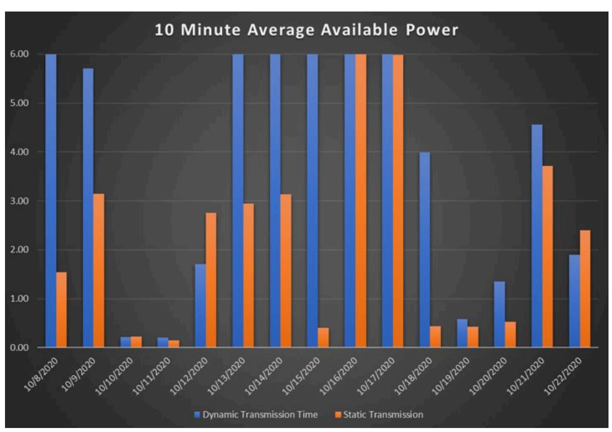

Fig. 4: 10 Minute Average Available Power

Conclusion and Future Work

Based on the findings of the model comparing the static event time versus a dynamic event time, the dynamic event is proven to be a better method in terms of mitigating the power demand. However, there are still many improvements that can be implemented in further development. The system-on-chip and microcontrollers used here were commonly available development boards that are designed to be used in a wide variety of applications. The RPi has features that would simply not be needed in the final product, such as USB ports, Ethernet/Bluetooth modules, and HDMI outputs. Although these features can be turned off in the settings of the OS, developing a single hardware system capable of handling the computational and control requirements of the device without the unnecessary features would save on cost and power demand.

Replacing the servo motors with stepper motors is another improvement that could be implemented. Servo motors rely on pulse width modulation to control the speed at which the motor runs, making positional data difficult or impossible to record. Stepper motors on the other hand can easily determine step counts and angles. The addition of sun angle to the collected data could be an additional parameter to use for predictions, and correlations between environmental factors and sun angle can be shown by the HMI.

Finally, improvements to data handling methods could be implemented to reduce or eliminate the need for human interaction. Older, irrelevant data can be automatically deleted from the transmitters limited memory space. The device currently needs maintenance every 150 days, but automatically deleting old data out in a FILO method would remove the need for that routine maintenance. More ambitiously, further developments could implement a self-analysis algorithm for accuracy. Variables like how far in the past to look, and how far into the future to predict could be adjusted by the device itself.

References

1. *Waste Prevention Information Exchange of ion due to graphite*. California Department of Resources Recycling and Recovery, https://www.calrecycle.ca.gov/ "Batteries". (accessed Dec 2020)
2. Z. Chen, “Designing Batteries for Easier Recycling Could Avert a Looming E-Waste Crisis”, https://www.mbtmag.com/home/news/21199734/designing-batteries-for-easier-recycling-could-avert-a-looming-ewaste-crisis, Oct 23, 2020, (accessed November 2020)
3. K. Smith, E. Wood, S. Santhanogopalan, G-H Kim, Y. Shi, A. Persaran, “Predictive Models of Li-ion Battery Lifetime” in IEEE Conference on Reliability Science for Advanced Materials and Devices, Colorado School of Mines, Golden, CO, September 7-9 2014
4. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016, http://www.deeplearningbook.org.
5. D. Madhubabu and A. Thakre, “Long-short term memory based channel prediction for SISO system,” in Proc. Int. Conf. Commun. Electronics Sys. (ICCES), 2019, pp. 1–5.
6. M. Abadi, et.al., “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” CoRR, vol. abs/1603.04467, 2016. [Online]. Available: http://arxiv.org/abs/1603.04467
7. “millis() [Time]”, https://www.arduino.cc/reference/en/language/functions/time/millis/, (accessed Dec 2020)
8. “HopeRF RFM69HCW”, https://www.hoperf.com/modules/rf_transceiver/RFM69HCW.html, (accessed Dec 2020)

© American Society for Engineering Education, 2021

Biographical Information

Imtiaz Ahmed

Dr. Imtiaz Ahmed is currently an assistant professor at Howard University and was at Marshall University until 2020. He works in the areas of wireless communications, signal processing, and computer networks. He completed his Ph.D. in Electrical and Computer Engineering (ECE) from the University of British Columbia (UBC) in 2014 and worked as a postdoctoral fellow at McGill University during 2014-15. Dr. Ahmed worked as a wireless systems engineer in Intel Corporation before joining Marshall University. Currently, he is working on artificial intelligence (AI) assisted physical layer design for 5G and beyond 5G cellular communication systems, UAV based communications, energy harvesting communications for massive IoT, etc.

Wook-Sung Yoo

Dr. Wook-Sung Yoo is a professor and chair of Computer Sciences and Electrical Engineering department at Marshall University. He has diverse academic background in Computer Science, Software Engineering, Computer and Electrical Engineering, Cybersecurity, Data Science, Health Informatics, Bioinformatics, Dentistry, and Dental Informatics. His current research interest includes artificial intelligence, machine learning, cybersecurity, data science, web technology.

© American Society for Engineering Education, 2021