Two-Cycle Renewable Energy Hybrid Power System

Alex Carpenter, Chris Cruz-Paez, Brian Dunsmore, Yonder Salomon

College of Engineering and Computer Science, University of Central Florida, Orlando, FL, 32826, USA

***Abstract* — This paper will present the methods used to design our two-cycle energy system consisting of solar and hydroelectric power as its sources. The purpose of this project is to serve as a proof of concept, showing that it is possible to have a hybrid renewable energy system which can allow a user who installs it to go completely off the grid and allow for a greener alternative to current power and electricity related challenges in the modern world. Alongside this, we will be showing our designed control systems which allow us to exercise a great deal of control over our power system, allowing it to feel very modern in terms of its design and implementation.**

***Index terms* — PCB, EER, LAMP architecture**

I. Introduction

Modern power systems are becoming smaller, more power efficient, and greener as time goes on. This is because the push for renewable energy is forcing governments, companies, and even consumers to act and contribute to the issue of global climate change. To contribute to the solution to this issue, our group developed this idea to make use of multiple renewable energy sources to be able to cover each other's weaknesses.

The two types of renewable energy sources we chose were the two we thought would be most practical in conjunction in residential homes and smaller scale applications, being solar power and hydroelectric power. With the combination of these two systems, we theorized we could have a continuous stream of power operating a load with the addition of a battery source. The expectations for our senior design project were to be able to generate a level of power from the system we calculated would be enough to sustain a constant power output to run a load continuously, while also being able to control the load and its sources wirelessly for a modern controls aspect.

This project was primarily focused on the implementation of the system and its components, more than we focused on the efficiency ratings of our systems. This is because at a smaller scale, such as the scale being performed in our senior design course, higher efficiency is harder to achieve with cheaper and smaller components. On a larger scale, renewable energy sources and power systems are shown to have higher overall efficiency ratings. This is most prevalent with our water pump, which will be the primary reason for our overall efficiency rating going down.

Through the implementation of a two-cycle power system, we are able to achieve a form of system which has an overall higher efficiency rating regardless of individual component efficiencies, due to the nature of power generation being inconsistent. With solar energy, there are many times during a natural day where the sun is being blocked or obscured by clouds, trees, planes, and other factors. This means that having a hydroelectric system in tandem with our solar system allows us to cover for the times when our solar system is unable to produce enough energy to sustain the system itself and allow us to achieve a much more consistent power generation flow.

Solar and hydroelectric energy are two types of energy which show promise as having the highest EER, or energy efficiency ratio. Others with high EERs are wind and biomass, but those forms of renewable energy generation were too bulky or impractical to implement on a smaller scale, regardless of efficiency rating. Solar shows an average efficiency rating of around 15% to 18%, while hydroelectric power shows the highest efficiency ratings of all forms of renewable energy, with the ability to convert about 90% of available energy into electricity.

II. Problem Formulation

The two primary problems we will be attempting to solve will be the issue of attempting to control a system wirelessly using built in control circuits on the PCB itself and using solar energy to power a system while trying to circumvent solar energies biggest weakness, the weather.

To begin, the wireless aspect of our system is one of the things we wanted to integrate into the system to make it more modernized. When attempting to control our system wirelessly, our first concern was with designing a control system that would allow us to give an input on the PCB controller itself, telling the system which source or load it should be using for whatever purpose we wanted. This was done using a Wi-Fi compatible chip and some inputs using a website of our choosing alongside system integrated controls, which will be discussed in further detail further in this document.

Another issue we have is the issue of wireless data transfer and displaying the data we collect from our system on a website. Due to the nature of our project, we will have a lot of data coming in from our system at any given point, giving us a big backlog of data that we need to process and sift through. We will be doing this using sensors integrated onto our PCB, along with code and a backend server to be able to track and save our data to be displayed later and will also give us the opportunity to check system data in the long term.

Looking at solar generation, the advantage comes in the large abundance of thermal energy provided by the sun. However, this means that only during certain times of the day can the highest amount of energy be generated. This system is also subject to the weather. If there is a cloudy or rainy day, the amount of energy that can be produced will be lower than on a bright sunny day. Another disadvantage is the time of the day. In the early morning and late afternoon hours, the amount of energy decreases.

During the initial testing of the solar panel, we were able to see this disadvantage by simply covering up parts of the panel. It was quickly noticed that the number of amperes generated would drop by large sums. However, voltage rating would decrease at a slower rate. It would take close to half of the panel covered to see a significant drop in voltage.

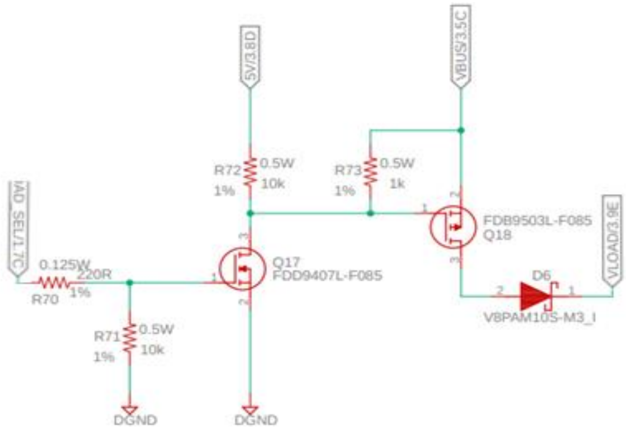
The second source of energy generation comes from the use of a micro hydro generator. Hydro generators can run for up to 21 hours a day. This allows them to produce large amounts of energy on a day-to-day basis. However, they require more maintenance than other forms of generation. Each generator needs to turn off for a cool down time. Otherwise, the system will overheat and cause damage to itself. Also, the system will require routine checks of the internal parts.

This type of preventative maintenance is essential to the long life of the system, but also lowers the amount of energy that can be created over a year. The quality of the water will affect the amount of energy that can be created. If the generator absorbs a large amount of air into the system, then the power output will decrease.

III. System overview

In order to implement a smart system with modern control by today's standards, a series of switches comprised by logic level FET based switches was implemented. Aside from discrete control of power sources, the system is also capable of gathering live metrics and transmit the data to a remote server featuring a LAMP (Linux, Apache, MySQL, PHP/Python/Perl) architecture. The data acquired is stored on a MySQL database and then displayed on the controls website for user analysis and input. For power sensing, INA260 power sensors by Texas Instruments were utilized. These are 16-bit precision, powerful, rated for 15A, sensors interfacing via I2C to the main MCU, the Raspberry Pi Pico.

As the "brains" of our system's operation, two microcontroller units were utilized in order to maximize control and sturdiness of the control system. The microcontrollers of choice were the RP2040, housed in the "Raspberry Pi Pico" developmental board and the ESP32 by EspressIF housed on a similar name board. The main goal of the ESP32 is to merely provide wireless connectivity to the system. The Raspberry Pi Pico houses all of the software responsible of producing telemetry data going into the remote server and to process telemetry data received from the server and actuate the requested interfaces. On the figure below an example of one of the logic level FET based high side switches is shown:



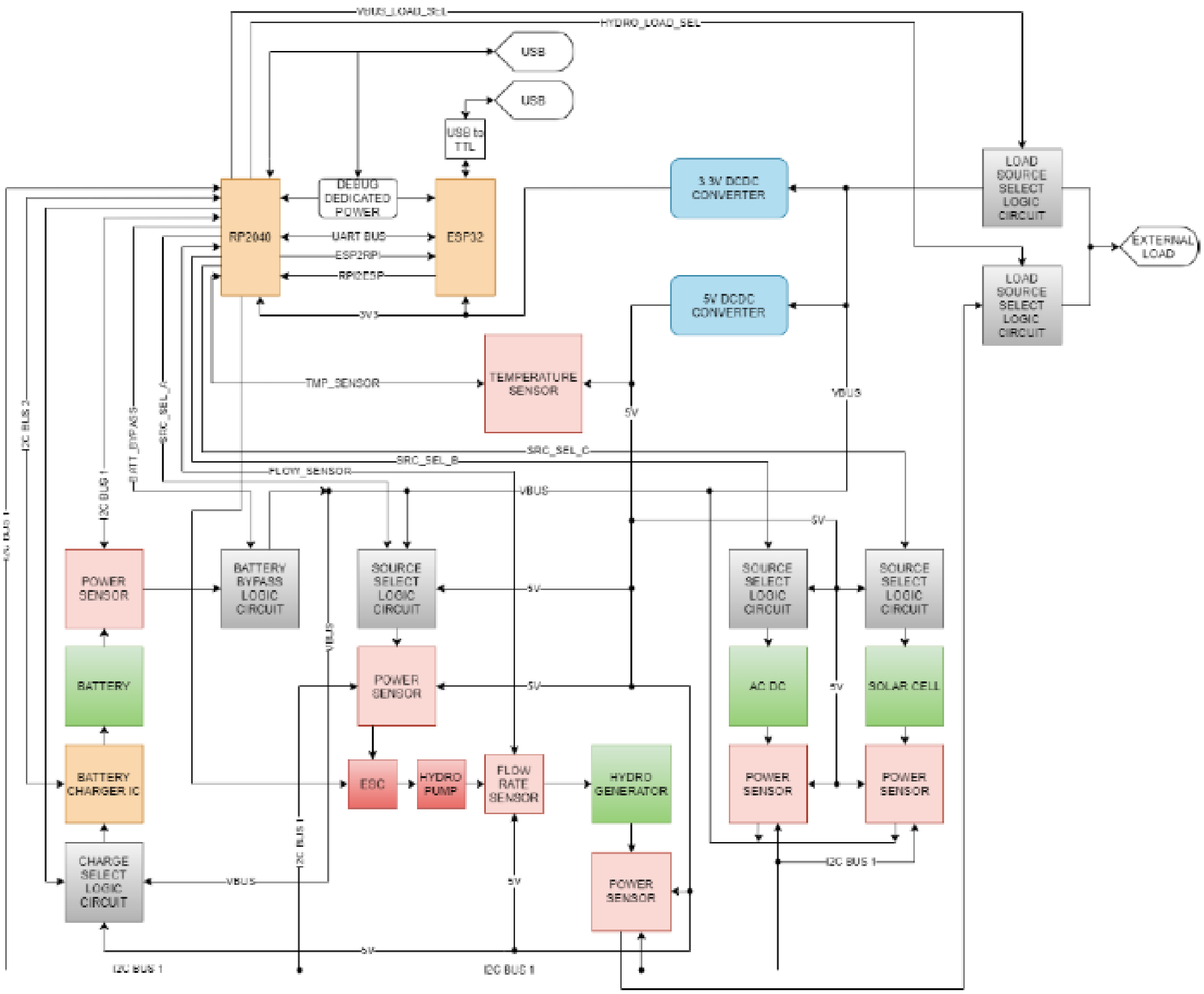
This circuit allows for a "high side" configuration for subsequent circuitry that needs a stable ground plane. Part of the possible areas of improvement and further exploitation to possibly make the system "smarter" include

- Implementation of Deep Learning algorithms on the server that can further analyze the data to provide metrics such as predictions in peak power output for each source, monetary savings from using the system among others.
- Automate power rails switching based on availability, eliminating the need for a user to manually select power sources. This feature could be implemented on both server side and system side.

To further describe the system, a common bus or net, named Vbus can be fed from either battery power, solar power, or AC 120V. This bus is utilized as the input for the system intrinsic power for basic operations, meaning the 3.3V and 5V voltage rails. This Vbus rail is also gated through the system to potentially be selected as a load feeding source. The only power sources that feed an internal 16V buck boost converter are the AC input and the solar cell. The reason for this is because the battery charger input needs at least ~2V over the battery float voltage in order to efficiently charge the battery. The hydroelectric power generated is gated through the system but can only be selected to be the source of the output for the external load.

For the physical components of our system we have the solar panel, hydro pumps, battery, and project housings. These components will be used as our primary forms of power generation and housing. The microcontrollers will have the ability to swap between using different sources of power as the primary source of power generation and will also be able to swap between different load sources for our power generated to be distributed to. These tasks will be handled by separate PCBs, which is the reason we have a PCB for power generation and delivery alongside a PCB for sensor data and controls.

The two PCBs alongside the battery for our system will be housed inside of a custom-made housing. This housing is going to be comprised of a combination of handmade wooden panels, and a 3D printed lid which will act as a removable lid for our system to show the PCBs and battery system. A generalized flowchart which consists of our systems key components can be seen in the figure below:



IV. PCB Controllers

For this project, we developed and designed two individual PCB controllers, as stated previously. We had originally designed a singular PCB controller that would be able to handle all of the computation, control, and power aspects of our system, but this proved to be difficult to produce correctly. Though we had simulated our PCB many times, we had received a PCB controller that had all of our 3.3V circuitry grounded, which we believed to be a manufacturing defect, as all of the PCB blanks we had been sent had this issue.

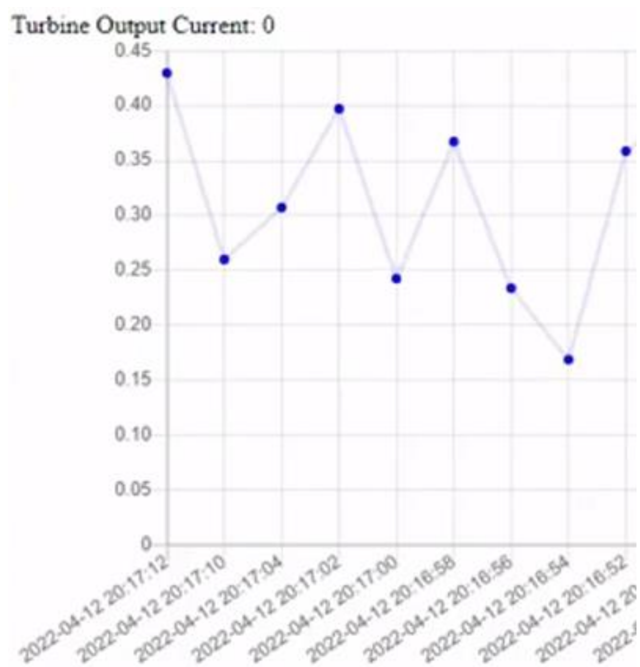
This part of our project was one of the most substantial portions of our project, as without both of these controllers we would not be able to properly control the system, nor would we be able to pull data from the system to see what was working and how well it was performing.

One controller was responsible for the aspects of power in our system, including the 3.3V and 5V power routing from the primary chip to the other parts of the systems sensors, computers, and other components. This PCB was also responsible for controlling the transformation of power using an on-board transformer for the incoming power from the panels and outgoing power to the battery system. This was done by using a transformer on our PCB along with two large capacitances to handle the large amount of power transformation necessary for this part of the project.

Next, we have the PCB controller responsible for the communication and computation of our system. This PCB consists of sensors, our ESP32, and our Raspberry Pi Pico. Integrated onto this board is all of our sensors for the system such as our voltage, current, and power sensors, and other control related FET logic circuitry, such as the battery FET circuit shown earlier in the paper.

The complexity of the second micro-controller PCB comes in its wireless and control capabilities. This PCB is the one responsible for physically performing the changes in the source and load selections, as well as being responsible for turning our water pumps on and off. This capability was integrated through the use of a back-end server alongside code which was running off of our Raspberry Pi Pico.

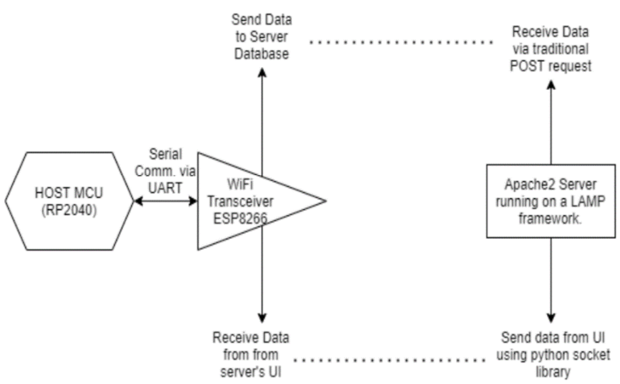
This board also had the capability to wirelessly transmit our data using this combination of things and allowed us to send packs of data to a web server we had hosted. We would then take this data and turn it into useable data that could be posted into a graph for nicer presentation, as shown in a snippet of our graphs below.



All of the programming for this project was done in python, using Thonny as our IDE to make the process of transmitting our data possible. Thonny forces the server to make a post request to the Wi-Fi module, which then exports the sensor values stored in variables to the server on a repetitive basis. The web server and website were made using PHP.

V. Software

In terms of system software for our project, it was mostly based on the LAMP architecture previously mentioned. Through this architecture we were able to find ways to wirelessly transmit data and control our system wirelessly as well, thanks to all the integration done on the sensor controller described in the PCB section. Located below is the basic architecture for the communication protocol between the remote server and the ESP8266 Wi-fi Transceiver.



The remote server is an Apache2 server running on the LAMP architecture mentioned previously. The server simultaneously sends user interface data to the ESP8266 while receiving system data via a traditional POST request. The user interface data sent from the server is received by the ESP8266 and is delivered to the RP2040 where the data is then used to interact with and change the parameters of the system, including changing the parameters of the pump control, energy source selection, load selection, as well as the battery charge selection.

The central methodology for implementing a working user interface on the system was storing each of the user inputs available on the server into a single integer array, which is sent to the ESP8266. This integer array once delivered to the RP2040 allows the MCU to alter the system to fit the parameters chosen by the user. Located below is an example of how the RP2040 interprets the data received from the server user interface.

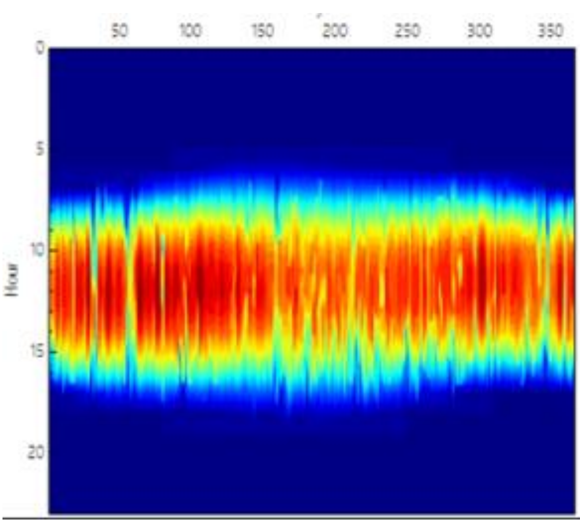
```
def uart_read(pin):
    try:
        utime.sleep(0.05)
        if uart_esp.any():
            data = uart_esp.read()
            data_dec = data.decode()
            #print(data_dec)
            x = json.loads(data_dec)
            print(x)
            if x[0] == '1':
                if [1] == '0':
                    hydro_load_sel.value(1)
                    vbus_load_sel.value(0)
            if x[0] == '0':
                hydro_load_sel.value(0)
```

This example from the main.py python script located on the RP2040 shows the effect the user has on the behavior of the system. Upon receiving any change in the system via a discrepancy between the current system status integer array and the new status integer array, a series of if statements are utilized to determine how the system should respond. In the present example, if the first element of the integer array is “1” and the second element is “0”, the hydro-electric pump is enabled and the Vbus load is disabled. Likewise, if the first integer array element is “0”, the hydro-electric pump is disabled. This methodology is applied to every available user input, with appropriate system responses based upon the values registered in the integer array.

In addition to the capability for the user to send user input data to the system, the system likewise sends data back to the server for system monitoring. This capability is achieved by the server making a POST request to the ESP8266, which will then send system data in the form of individual variables such as battery voltage, battery current, AC input voltage, solar cell output voltage, etc. Each of these individual variables in independently m as functions of time, the plots of which are located on respective pages of the server website.

VI. Solutions

When looking at how we chose to solve the issues we discussed earlier, we have two primary issues and the steps we took to solve them. As stated earlier, we found early on in the systems testing that the solar panel was very sensitive to changes in weather, specifically sensitive to blockage of direct light. This image was created by using the program PVwatts. We were able to input weather patterns over the course of a year and obtain the number of watts that could be produced by the panel of choice. The image below demonstrates the time of day at which our solar panel will be at its peak performance.



As we can see in the image provided above, for solar the window of generation is limited to the time of day and weather. However, hydro generation can be used in both cloudy days and twilight time of day. In our design we implemented a system that will produce energy, from solar, during the peak hours of the day. Using a single solar panel, we were able to produce a sustainable 10 Watts of power from the solar panels only.

However, this is only available during the hours of 9 am to 3 pm. After this time the amount of power the panel can produce drops out of the designs range. To keep the system maintained at a 10-Watt value, we had the system switch to a hydroelectric system. This process was done remotely from a designed website as discussed earlier. Here we can continue to supply power to the load and keep the supplied power at the required range.

Using our systems functionality to send data to the website in real time, we were able to gather testing data which showed us the voltage and current from the solar panels in real time. During peak sunlight, the power generation would be higher than the numbers presented here, getting about 22W during peak times when testing at other points.

Below, we have an image that shows the output we were getting from the solar panel at the peak generation from a time when we tested the panel.

QL Search Insert Export I

date	solar_voltage	solar_current
2022-04-12 18:40:28	17.01	0.49875
2022-04-12 18:40:30	18.335	0.55375
2022-04-12 18:40:32	15.8088	0.59
2022-04-12 18:40:38	18.6725	0.52375

With this data, we can see that the system will produce the required amount of wattage and be able to switch between voltage sources without having a significant drop in power generation. Therefore, we can conclude that the proposed solution was a success in solving the problem.

As for the wireless controls section of our project, we were able to solve this problem using a method of interfacing between the system, code, and a remote server, which allowed us to be able to control the system from any distance, given we had access to the wireless server. This gave us the ability to enable and disable loads at will using control systems we designed on our PCBs ourselves.

VII. Conclusions

Looking back at our project, the struggles, and solutions, this project was incredibly valuable to us. Having the opportunity and liberty to dream of a system that interested us and begin to actualize the system and bring it to life was a great experience for all of us. This coupled with the opportunity to experience how the process of designing and developing a project in a group, where everyone had set tasks, deadlines, and communication aspects they had to accomplish was very valuable to us.

Whether it was contributing ideas to the group, getting along outside of project hours, or clashing ideas, we believe the experiences gained from our senior design project were experiences that will help us become better engineers and people in general. It is important to learn how to co-operate, but also important to learn how to trust others to do their part of a project and depend on others while working together.

It goes without saying that senior design was a difficult course for all of us in the group but was a very realistic representation of the stresses of real-world project engineering and the tight deadlines that can come with it. The experiences of our four group members coming together really made this senior design project of ours feasible, as we all contributed a different breadth of knowledge to the project, most of which would have been difficult to learn all on our own if we had to. All in all, we believe we were able to accomplish everything we wanted to from this senior design project, and much more. We were able to accomplish stretch goals, and we are grateful for the opportunity to have been able to work together.

VIII. Biography

Brian Dunsmore is an Electrical Engineering major graduating in Spring 2022. He chose to pursue the Power and Renewable energy track for this degree program. After graduation, he plans on working for Whiting-Turner as a MEP Project Manager. He also plans on obtaining his PE within 3 to 5 years.

Alexander Carpenter is an Electrical Engineering major graduating in Spring 2022. Since November of 2021 he has been working with Athena-Tek building LTE and 5G capabilities as well as mesh radio networks. After graduation, he plans to work further with Athena-Tek as a full-time engineer, and later pursue his masters within 1 to 2 years following graduation.

Christian Cruz Paez is an Electrical engineering major, graduating in spring 2022. He found his focus on electrical engineering to be on circuit design at his current jobs. He plans to someday attend graduate school, while shifting his focus into a more Computer Engineering oriented degree path.

Yonder Salomon is an electrical engineering major, also graduating in Spring 2022. He is currently working for ProActive Technologies as a systems engineer. With his degree in the Power and Renewable energy track, he plans to soon take and pass the FE exam and later become a PE, along with pursuing a masters degree about a year after university.

IX. References

"Product". Topsolar-Energy.Com, 2021, https://www.topsolar-energy.com/product_view.asp?id=363.

Topsolar-Energy.Com, 2021, https://www.topsolar-energy.com/upimages/20179251056602827.pdf.

https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf

https://www.ti.com/lit/ds/symlink/tlv3501.pdf?ts=1635821530774&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTLV3501

https://www.electronicshub.org/classification-of-relays/

https://amperite.com/blog/relays/

https://components.omron.com/relay-basics/basic

https://www.sparkfun.com/datasheets/Components/LM7805.pdf

https://en.wikipedia.org/wiki/Voltage_regulator

https://www.digikey.com/en/maker/blogs/2020/what-is-a-voltage-regulator

https://isocpp.org/wiki/faq/coding-standards

https://en.wikipedia.org/wiki/Technical_documentation

https://www.iso.org/ics/01.110/x/
https://spectrum.ieee.org/boschs-smart-virtual-visor-tracks-sun

https://en.wikipedia.org/wiki/Liquid-crystal_display

https://www.ecmag.com/section/codes-standards/pv-installation-codes-and-standards

https://en.wikipedia.org/wiki/Comparison_of_wireless_data_standards

https://en.wikipedia.org/wiki/IEEE_802.11

https://www.espressif.com/en/products/socs/esp32

https://www.technology.pitt.edu/help-desk/how-to-documents/wireless-network-standard