4.3 Proposed Design

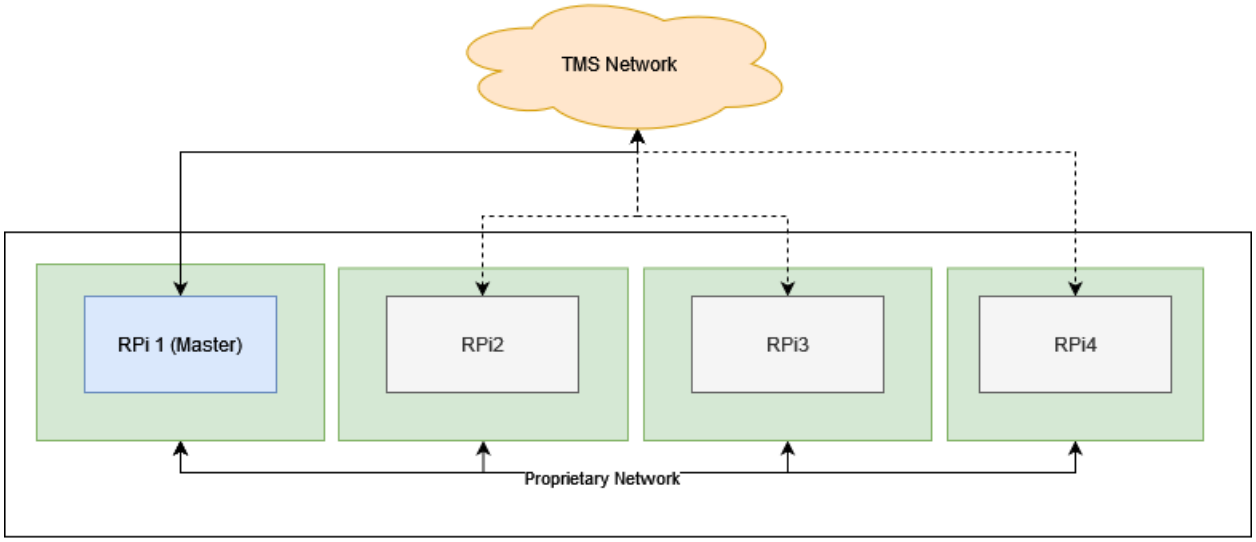
4.3.1 Overview

Provide a high-level description of your current design. This description should be understandable to non-engineers (i.e., the general public). Describe key components or sub-systems and how they contribute to the overall design. You may wish to include a basic block diagram, infographic, or other visual to help communicate the overall design.

The microgrid pallet communication will be controlled by a Raspberry Pi (RPi) device mounted on each microgrid. This RPi accepts commands from a connected network according to the Tactical Microgrid Standard (TMS) and translates them into messages specific to the microgrid unit. These messages are then forwarded to the microgrid array.

The components used for each microgrid pallet requires communication over a proprietary network to synchronize properly, so the entire microgrid array is configured as a single unit with one master RPi accepting TMS messages.

The Raspberry Pi additionally maintains an attached user interface for viewing the status and updating configurations for a microgrid. In this way the microgrid array can be controlled via network or an on-unit display.



Microgrid units (green) communicating between each other and the TMS network

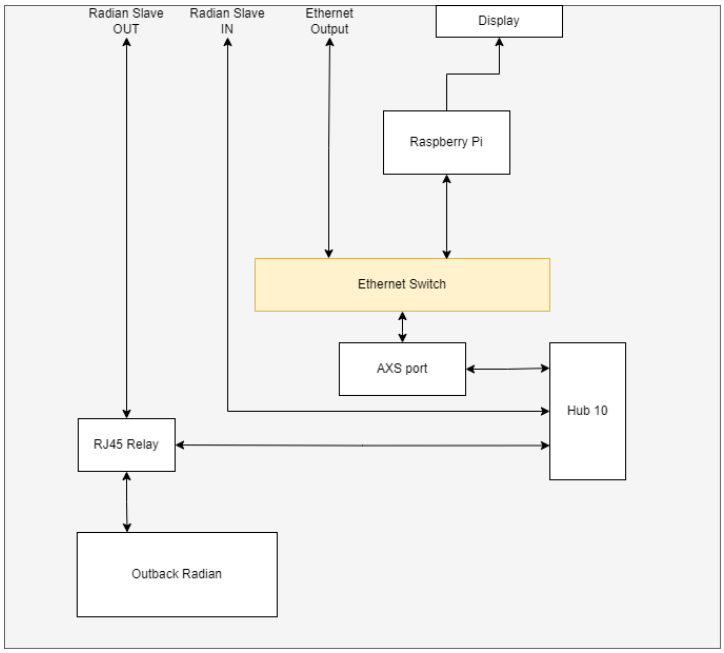
4.3.2 Detailed Design and Visual(s)

Provide a detailed, technical description of your design, aided by visualizations. This description should be understandable to peer engineers. In other words, it should be clearly written and sufficiently detail such that another senior design team can look through it and implement it.

The description should include a high-level overview written for peer engineers. This should list all sub-systems or components, their role in the whole system, and how they will be integrated or

interconnected. A visual should accompany this description. Typically, a detailed block diagram will suffice, but other visual forms can be acceptable.

The description should also include more specific descriptions of sub-systems and components (e.g., their internal operations). Once again, a good rule of thumb is: could another engineer with similar expertise build the component/sub-system based on your description? Use visualizations to support your descriptions. Different visual types may be relevant to different types of projects, components, or subsystems. You may include, but are not limited to: block diagrams, circuit diagrams, sketches/pictures of physical components and their operation, wireframes, etc.

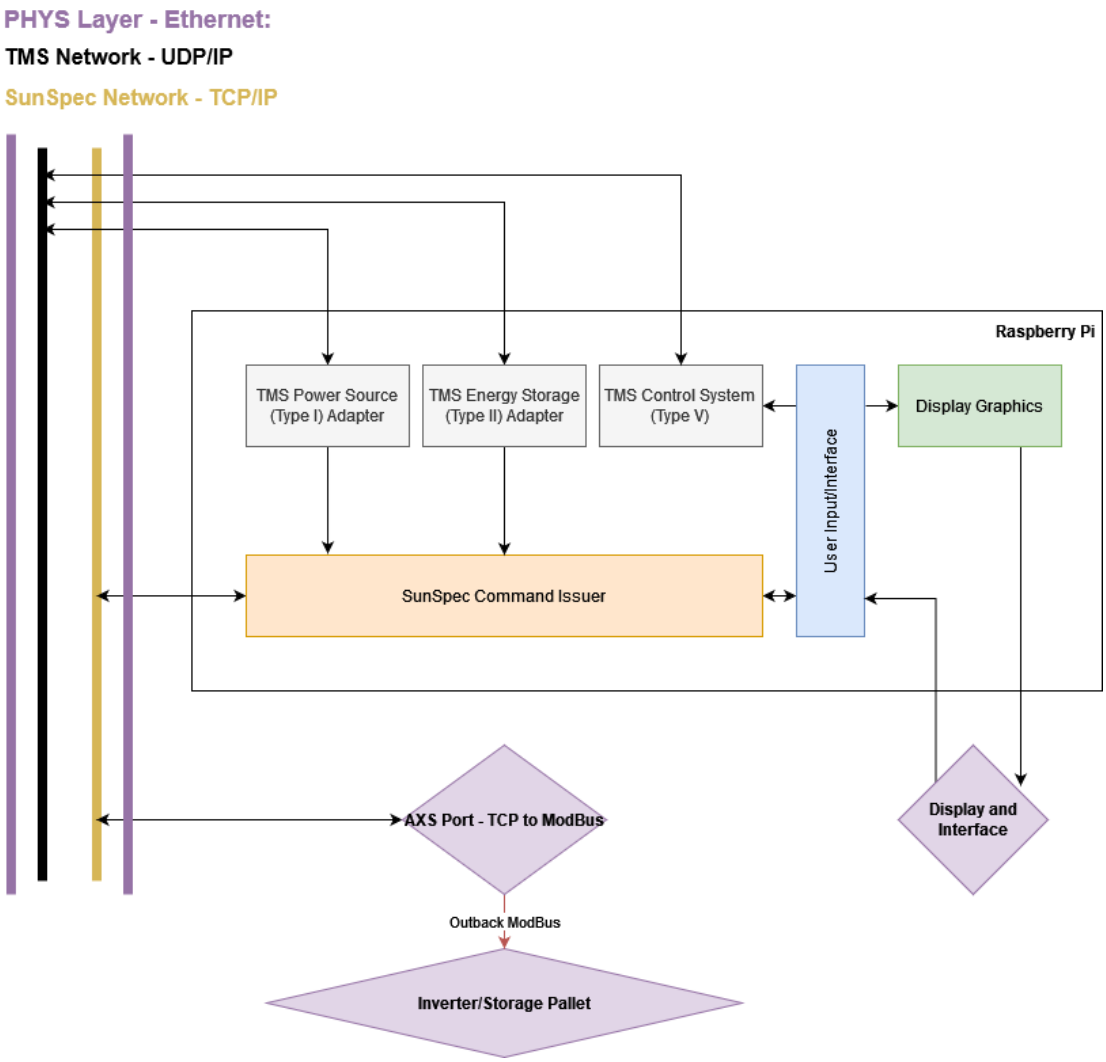


Above is a diagram describing our current plan for hooking up the Raspberry Pi to the rest of the Microgrid system. This diagram only concerns parts connected to and communicating with the Raspberry Pi and ignores the rest of the system. The Raspberry Pi will be connected to an Ethernet Switch which will allow it to communicate with the rest of the Microgrid and Microgrid system.

The Raspberry Pi will give commands to the Microgrid it is a part of through the use of the AXS port. The AXS port translates our pysunspec2 API code into Outback's proprietary communication protocol to allow the Raspberry Pi to communicate with the rest of the system.

The Ethernet Output will connect the Raspberry Pi to the other network of Raspberry Pi's the connect each Microgrid together. Through this, the master Microgrid's Raspberry Pi will communicate with the others on what they must do while the non-master Microgrids' Raspberry Pis will reply with their current status using heartbeat signals.

The display will show information about the microgrid that the display is attached to. The display will be coupled with some controls to allow the user to navigate through the menus on screen.



Above is a more software focused design of what will be running on each Raspberry Pi.

We have split the software into multiple sections to allow for each group member to contribute and work independently of each other. The TMS Power Source Adapter, TMS Energy Storage Adapter, and TMS Control System are all protocols that we must follow when designing this software. The purpose of these protocols is to standardize and meet powergrid system needs.

For the Display Graphics, we have been putting research into various python graphics libraries we could use. The one we are most likely going to use is GUIzero as it is simple and easy to implement which is important for us since most of us have not created applications like this before.

For the SunSpec Command Issuer, we will be using the pysunspec2 API inorder to send and receive data from the Outback equipment.

4.3.3 Functionality

Describe how your design is intended to operate in its user and/or real-world context. What would a user do? How would the device/system/etc. respond? This description can be supplemented by a visual, such as a timeline, storyboard, or sketch.

Our design is intended to reduce the amount of user input and technical knowledge needed to operate the microgrids both as a single unit and as an array of units. With our design a user would only need to select how they want to use the microgrid from the options shown on the display and plug in cables connecting the microgrid units for the microgrids to operate as needed. After receiving the user inputs the Raspberry Pi will send signals to the microgrid using the SunSpecAPI to have it start using the selected mode.

4.3.4 Areas of Concern and Development

How well does/will the current design satisfy requirements and meet user needs?

Based on your current design, what are your primary concerns for delivering a product/system that addresses requirements and meets user and client needs?

What are your immediate plans for developing the solution to address those concerns? What questions do you have for clients, TAs, and faculty advisers?

The current design satisfies our functional requirements for how the microgrid array should function.

One of our largest concerns is creating a user interface that is easy to use and see vital information about the microgrid system. We haven't spent much time designing this interface yet, and it will be an important factor in terms of the usability of our final product.

Another concern we have yet to determine is how well the capabilities of the Radian inverter/charger match up to the functions defined in the TMS. If not enough functionality is available, it will not be a proper implementation of the TMS.

The next step for our design and implementation work is to create translation functions between TMS commands and SunSpec messages to the hardware that will be changing register values. In addition, we will begin to design a GUI and input system based on the capabilities of the Radian and begin development with a Python graphics library to be selected.

4.4 Technology Considerations

Describe the distinct technologies you are using in your design. Highlight the strengths, weakness, and trade-offs made in technology available. Discuss possible solutions and design alternatives.

Technology	Pros	Cons	Trade-Offs
OMG DDS	Secure International Standard	We have no experience with this technology	
UDP/TCP/IP Networking	International Standard with lots of documentation and examples.		
SunSpec API	Allows duse so send commands to the Outback Power Hardware	We have no experience with this technology.	

	Open Source Test software available.		
Python	Fast development, easy to use. Available libraries for SunSpec messaging.	Slower execution than another language like C.	
TMS	We will be one of the first teams implementing this standard.	Not a lot of documentation since the standard is in development.	
Outback Power Hardware			The hardware requires a proprietary communication network, but it also handles complex synchronization problems such as phase matching that our team has no knowledge in.

4.5 Design Analysis

Discuss what you have done so far, i.e., what have you built, implemented, or tested? Did your proposed design from 4.3 work? Why or why not? Based on what has worked or not worked (e.g., what you have or haven't been able to build, what functioned as expected or not), what plans do you have for future design and implementation work? For example, are there implications for the overall feasibility of your design or have you just experienced build issues?

So far we have attempted to send SunSpec messages from the RPi to the Outback Power AXS port and retrieve the status on the Radian inverter.

The next step for our design and implementation work is to create translation functions between TMS commands and SunSpec messages to the hardware that will be changing register values. In addition, we will begin to design a GUI and input system based on the capabilities of the Radian and begin development with a Python graphics library to be selected.