Making Microgrids Plug & Play

Team: 36

Andrew Frank, Austin Thoreson, Christian Pinta, Ben Eder, Saketh Jonnadula

Client: Nick David Advisor: Mathew Wymore

Introduction

Problem Statement:

Microgrid pallets provide a mobile, modular power solution that can be applied to many use cases. Currently, connecting these pallets is time-consuming and requires technical expertise specific to these devices.

Solution:

Our goal is to simplify the connection and configuration of pallets such that minimal to no training at all is required to deploy and configure them.

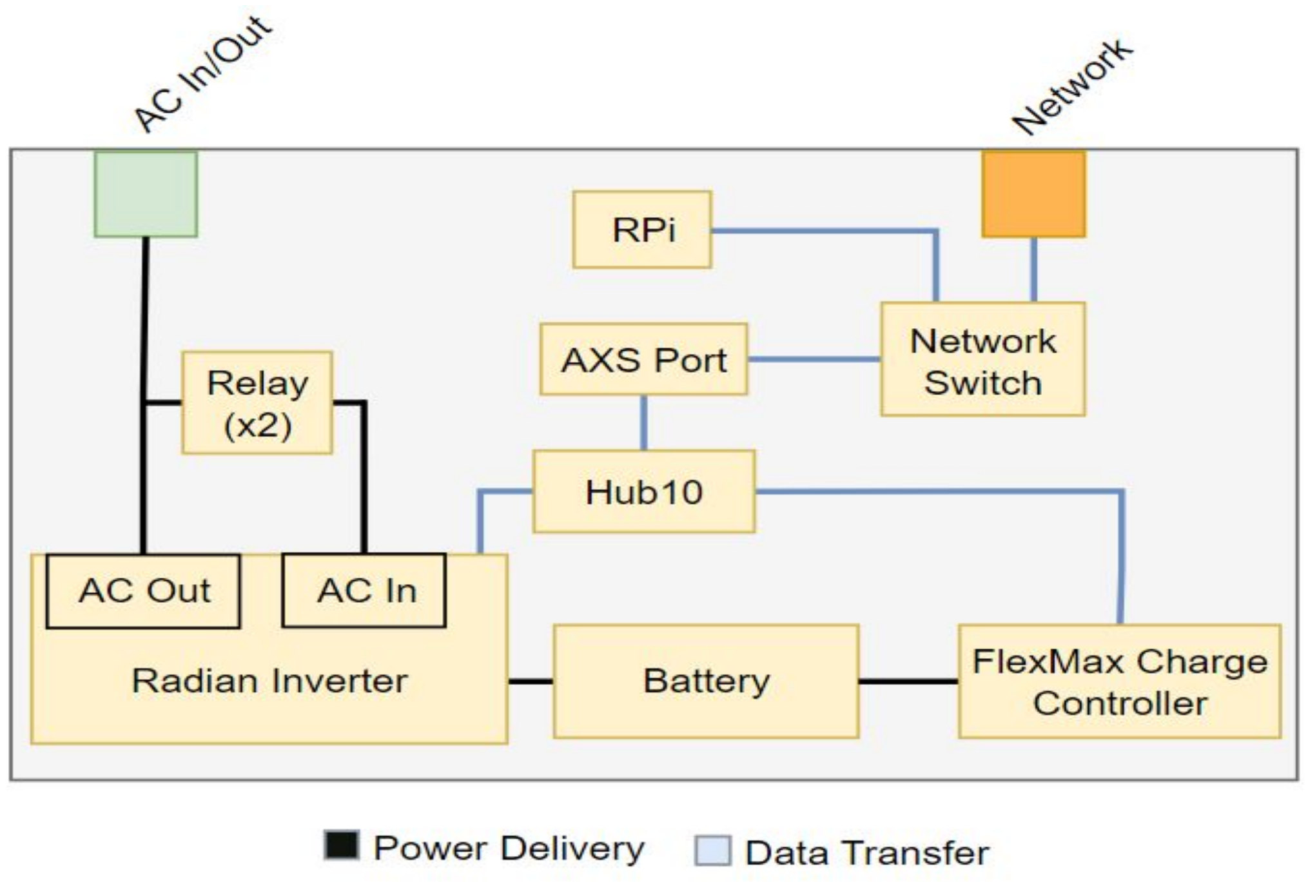
Impact

These Microgrids have the ability to deliver a medium amount of power for a long-duration (10+ hours). Simplifying their setup would allow them to be more easily deployed in a multitude of applications including disaster relief, pop-up carnivals, and potentially even military applications.

Overview

What is a Microgrid?

Microgrids are a collection of devices designed to store and deliver a medium amount of power (8-80 kW) in a package around the size of standard pallet. Each pallet can deliver up to 7.4kW



The devices used to construct these pallets are traditionally designed for a static, semi-permanent installation and thus are difficult to configure in dynamic deployment scenarios.

Reason for Config. Difficulties Include

- Physical connections change based on users choice of a leader/follower
- User interface devices are complex and require advanced system knowledge

Methodology

Design Approach

Do what we can to eliminate the configuration complexities.

We opted to try...

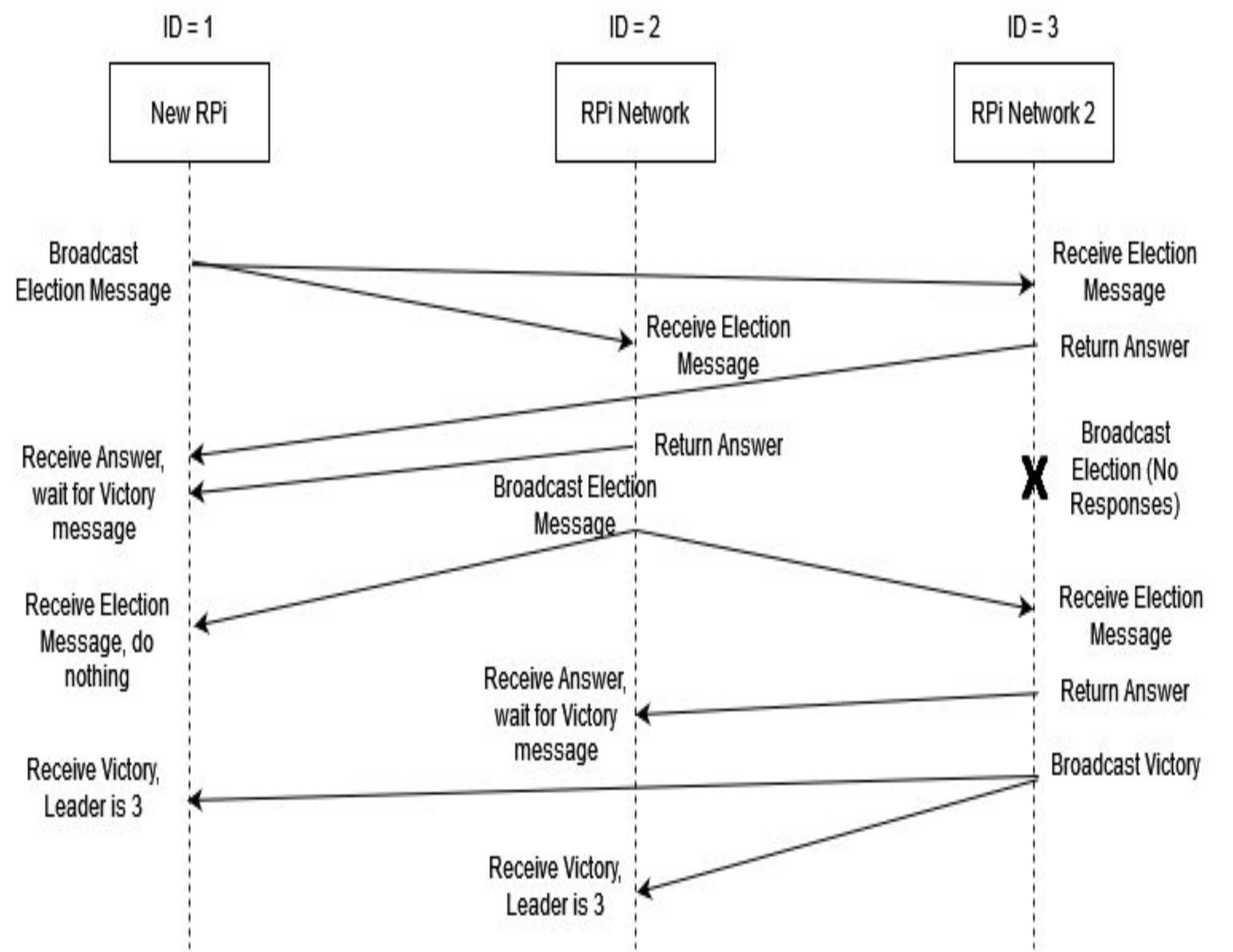
1. Electing a leader, taking it out of the hands of the user
2. Automating configuration changes via a Raspberry Pi microcontroller
3. Replacing the static proprietary network, allowing for more dynamic connections

After following this approach, we found we would also have to tackle AC source synchronization

Implementation

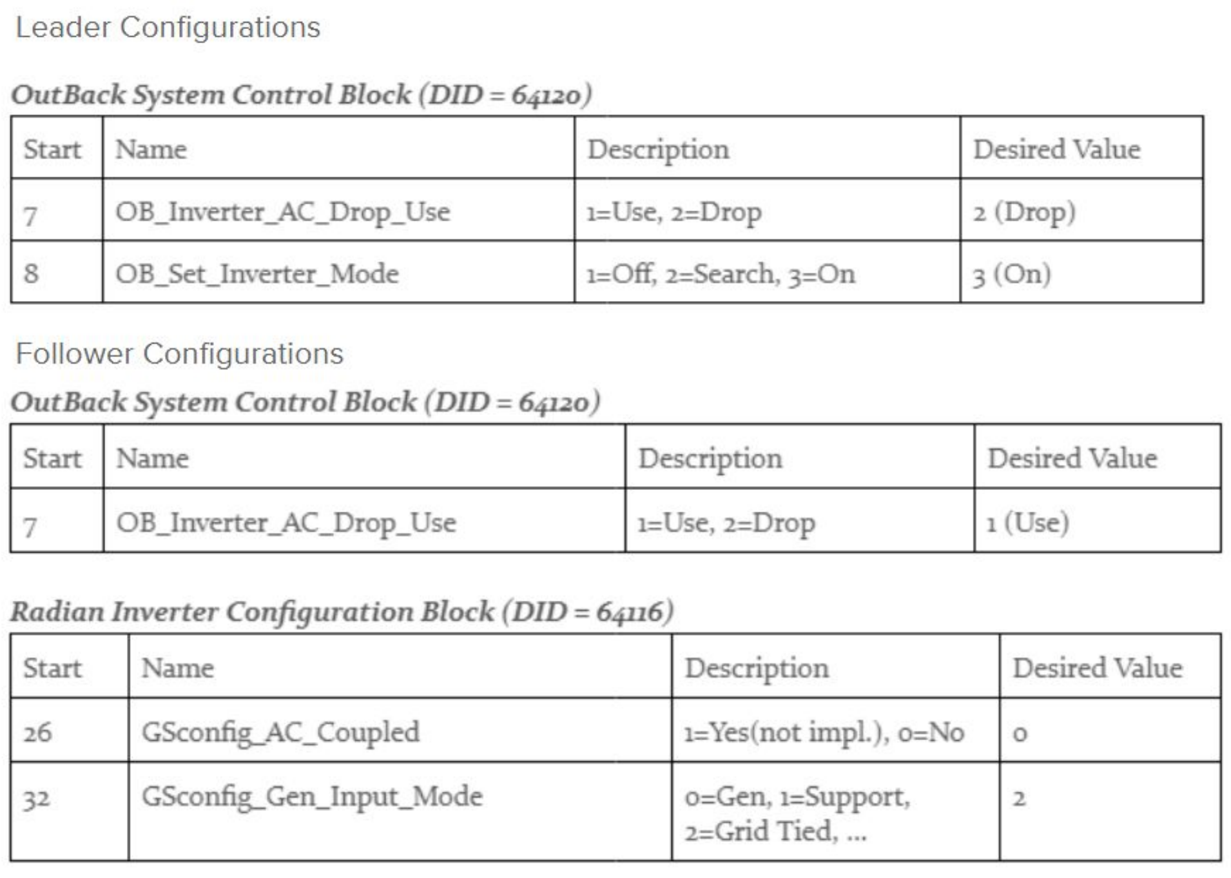
Leader Election

A series of messages allows distributed devices to reach consensus. The device with the highest ID becomes the leader.


ID = 1
ID = 2
ID = 3
New RPi
RPi Network
RPi Network 2
Broadcast Election Message
Receive Election Message
Receive Election Message
Return Answer
Return Answer
Receive Answer, wait for Victory message
Broadcast Election Message
X
Broadcast Election (No Responses)
Receive Election Message, do nothing
Receive Election Message
Return Answer
Receive Answer, wait for Victory message
Broadcast Victory
Receive Victory, Leader is 3
Receive Victory, Leader is 3

Driving Microgrid Configs

Functions to read, write, and verify data points in the Microgrid using the SunSpec API. A software state machine determines which points are set and in what order.



Leader Configurations

OutBack System Control Block (DID = 64120)

Start	Name	Description	Desired Value
7	OB_Inverter_AC_Drop_Use	1=Use, 2=Drop	2 (Drop)
8	OB_Set_Inverter_Mode	1=Off, 2=Search, 3=On	3 (On)

Follower Configurations

OutBack System Control Block (DID = 64120)

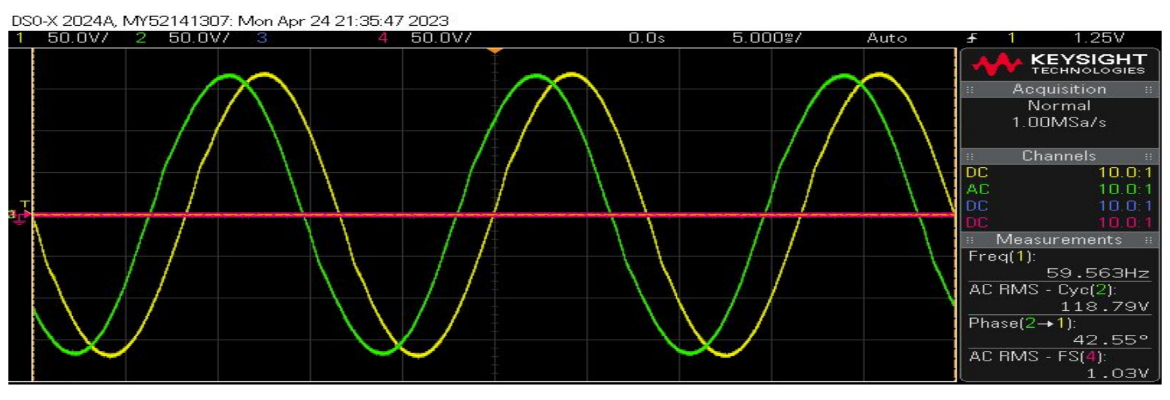
Start	Name	Description	Desired Value
7	OB_Inverter_AC_Drop_Use	1=Use, 2=Drop	1 (Use)

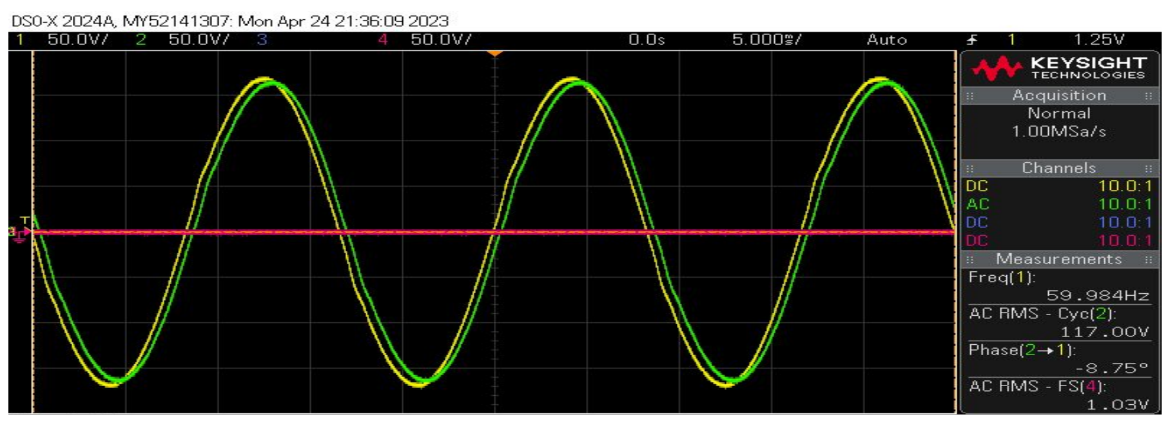
Radian Inverter Configuration Block (DID = 64116)

Start	Name	Description	Desired Value
26	GSconfig_AC_Coupled	1=Yes(not impl.), 0=No	0
32	GSconfig_Gen_Input_Mode	0=Gen, 1=Support, 2=Grid Tied, ...	2

AC Source Synchronization

Through the use of relays and some unconventional methods, we were able to achieve pseudo-synchronization across two microgrids.





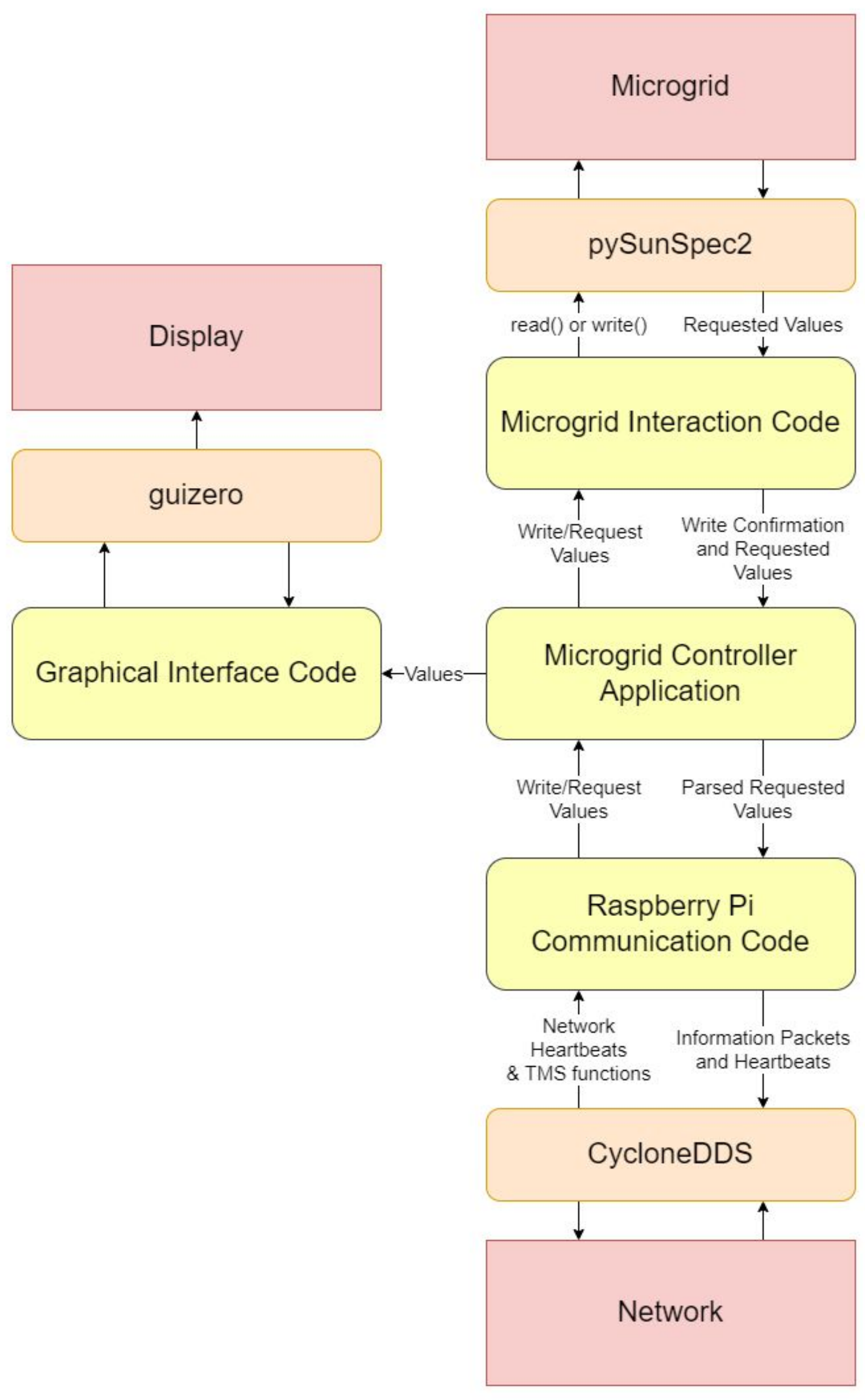
Results

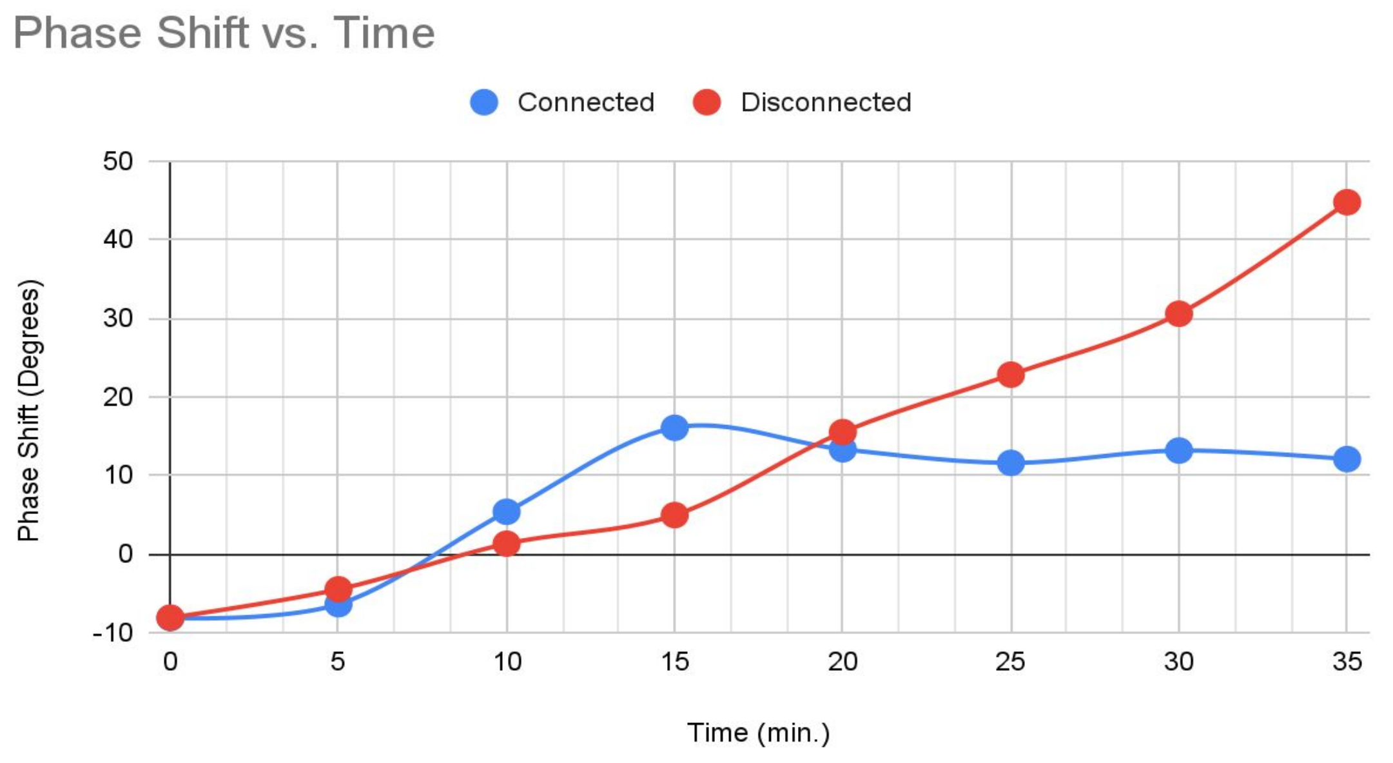
Software

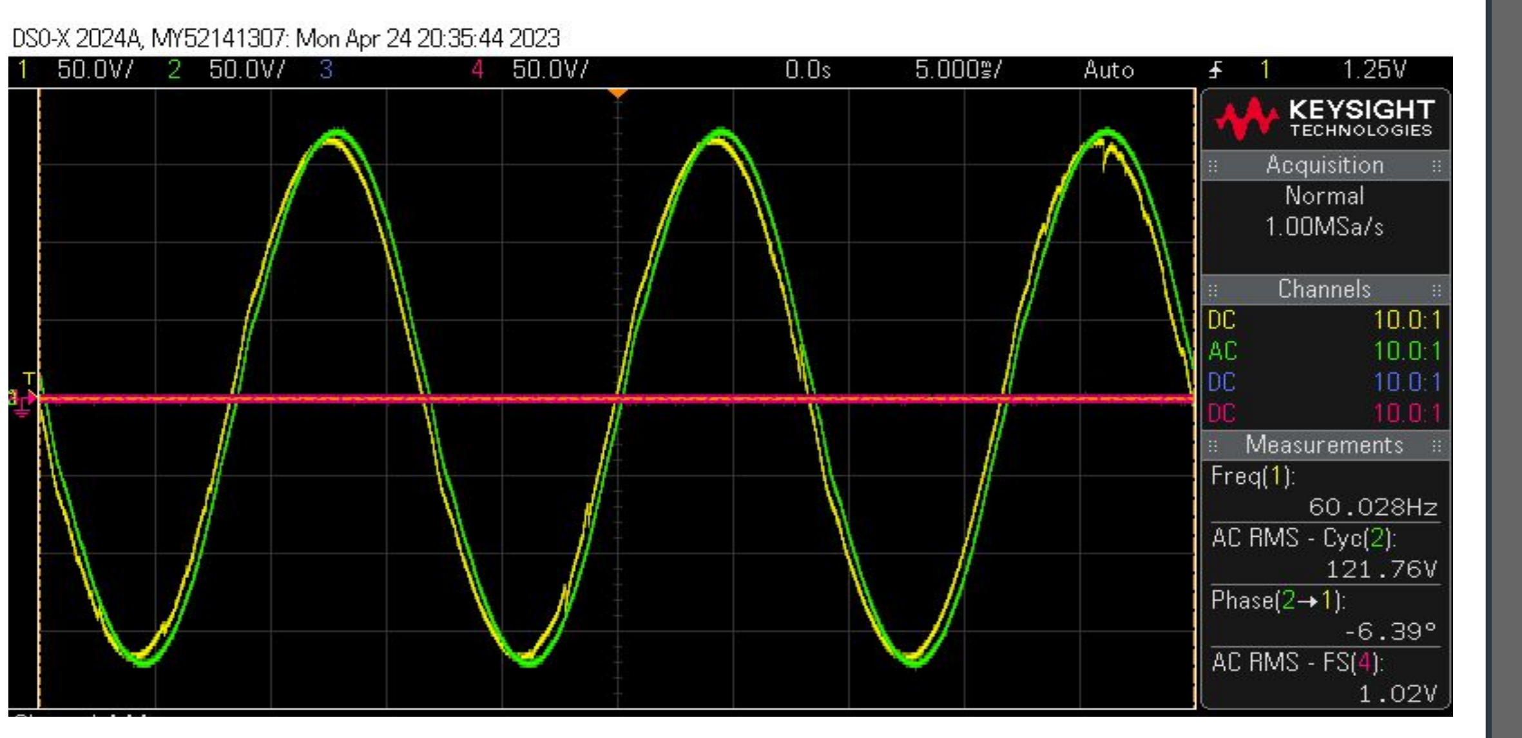
- Successful election and network event response
- Sensing and configuration for a pallet
- Mock pallet for software unit testing
- Unable to test on software due to unsolved communication errors

Hardware

- Significant amount of data acquired for various configurations
- Promising configuration process for phase alignment using grid tied modes
- Testing reliable and consistent methods for successful source synchronization


Microgrid
pySunSpec2
read() or write()
Requested Values
Microgrid Interaction Code
Write/Request Values
Write Confirmation and Requested Values
Microgrid Controller Application
Values
Display
guizero
Graphical Interface Code
Write/Request Values
Parsed Requested Values
Raspberry Pi Communication Code
Network Heartbeats & TMS functions
Information Packets and Heartbeats
CycloneDDS
Network





Conclusion

At the start of this project, the biggest question was if it's possible to link and automatically configure Microgrid devices. While we didn't reach an optimal final solution, our team has created base software for configuration, coordination, and viewing available data along with research on how the Microgrids respond to different configurations. We view this as a significant achievement.

Iowa State University sdmay23-36@iastate.edu