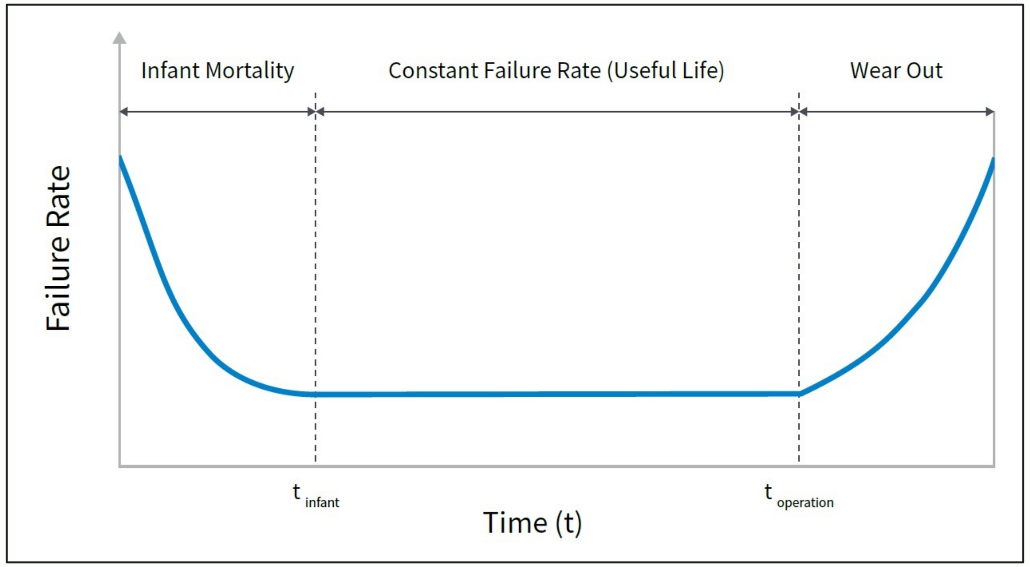
VFE Burn-in Setup Software

Christian Bernier (Northeastern University)
June 13, 2024

Overview

- Context
- Services
- GUI features
- Documentation

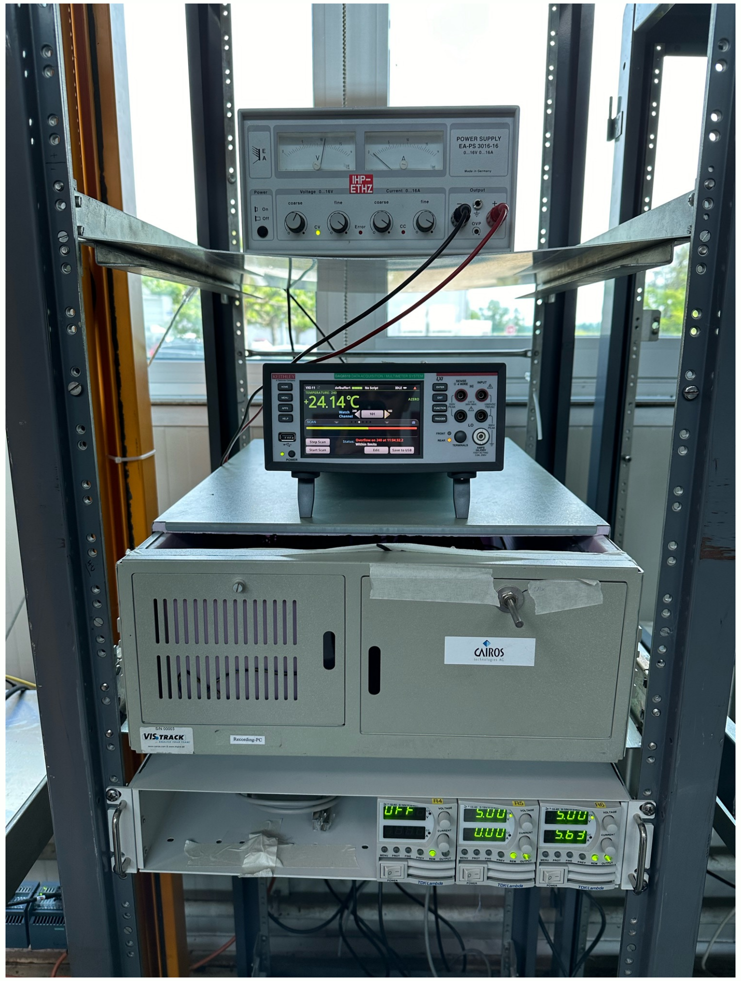
Context



- Many electronics follow a standard life cycle
- Running cards at 70°C for 1 week simulates 1 year of normal operation (accelerated aging)
- Goals:
  - Age VFE cards past the point of infant mortality before installation
  - Age many cards at once, so we can keep up with production

Hardware

- 3 burn-in racks, each with
  - 4 boxes
    - 45 VFE cards
    - Temperature sensors
    - Fans
  - 2 power supplies (2 boxes per PS)
- 1 control rack
  - PLC safety system
  - Control server running custom software

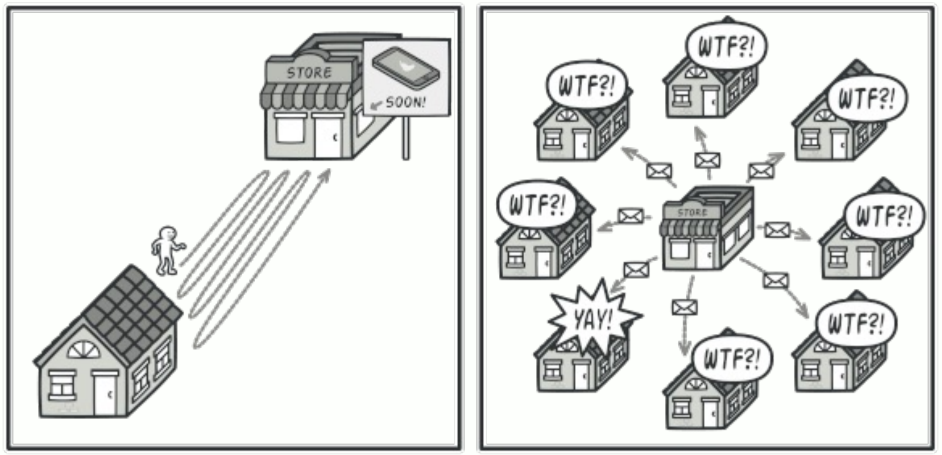


Services

- Each responsible for one aspect of the setup
- May manage drivers to interface with physical devices
- Can establish connections to other services
- Built using RPyC library

Observables/Observers

- Services or clients (observers) can register to be notified by another service (observable)
- Eliminates the need for polling services
- Only those who need the data receive it



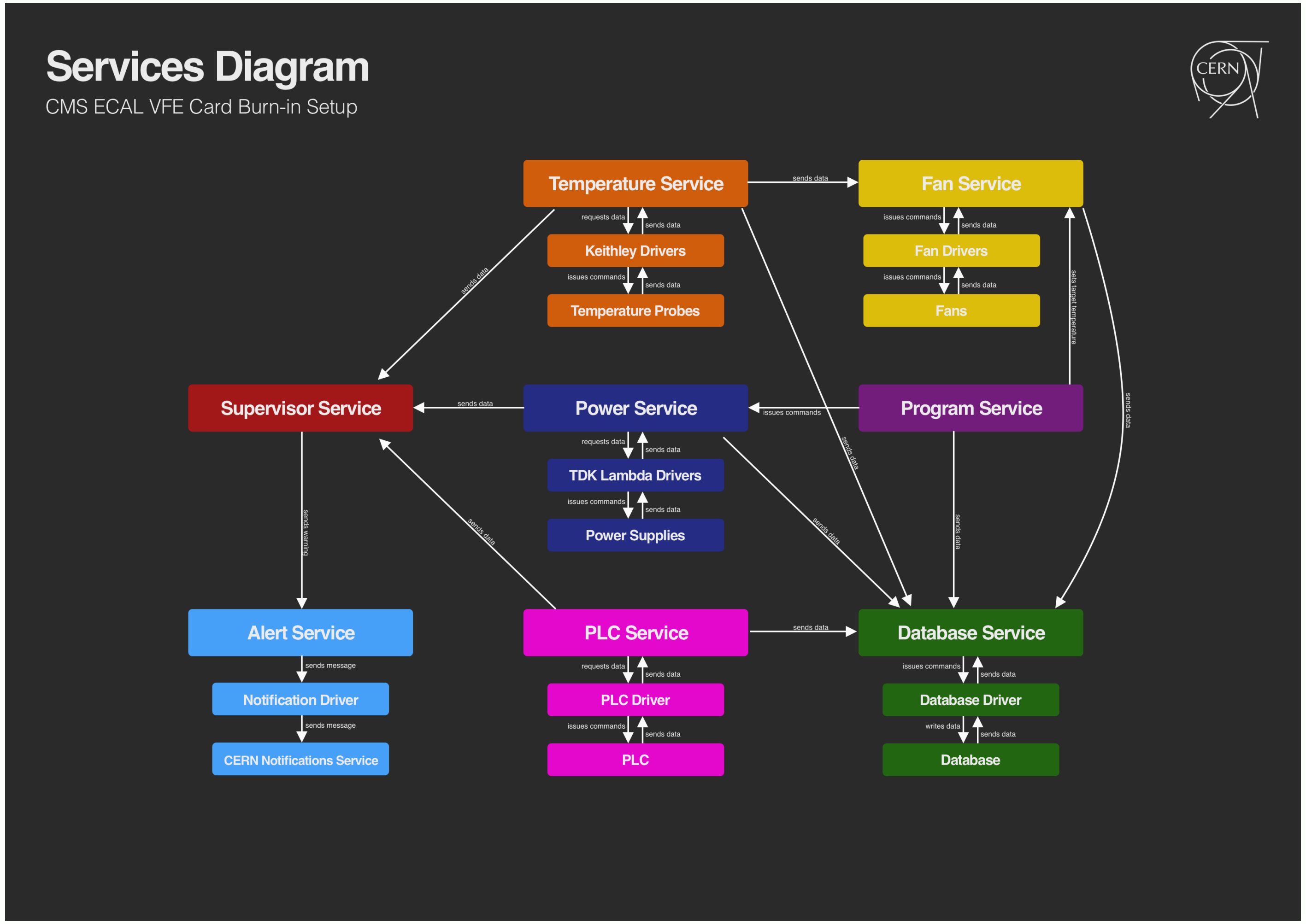
Before

The New York Times
THANKS! YOU ARE SUBSCRIBED!

After

Service Relationships

- Almost all services are observable
- Services measure data regardless if it is used
- Device-specific tasks are delegated to drivers
- All services are multithreaded to optimize performance



Temperature Service

- Observable for temperature data updates
- Uses driver to collect temperature data from EBKeithley devices
- Collects temperature data every 30 seconds
- If the device cannot connect:
  - Broadcasts an empty event to observers
  - Attempts to reconnect again in 5 seconds instead of 30 seconds

Power Service

- Observable for power data updates
- Uses driver to collect power data (voltage, current, interlock status) from TDK Lambda power supplies
- Collects power data every 30 seconds
- Just like the temperature service, if the device cannot connect:
  - Broadcasts an empty event to observers
  - Attempts to reconnect again in 5 seconds instead of 30 seconds
- Provides methods for turning power supplies on/off

Fan Service

- Observable for fan speed updates
- Connects to the temperature service to receive temperature updates
- Stores a target temperature range to maintain
- Utilizes a “fan strategy” which defines how to respond to a new temperature reading

Fan Service

- When a temperature is received:
  - Checks the current fan speed for each box
  - Asks the strategy for a new fan speed, given the current temperature and fan speed of each box
  - Records the new fan speed
  - Sends new fan speed to fan drivers to update the devices accordingly
- Driver still needs to be implemented

Program Service

- Provides interface for running automatic procedures on boxes
- Manages each program running on the setup, preventing multiple programs from running on the same box
- Observable for program status updates

Programs

```yaml
program:
  name: 'Program 1'
  run-on:
    - [true, true, false, false]   # Rack 1
    - [true, true, false, false]   # Rack 2
    - [false, false, false, false] # Rack 3
  steps:
    - type: 'TARGET_TEMP'
      min: 50
      max: 55
    - type: 'REPEAT'
      times: 30
      steps:
        - type: 'WAIT'
          seconds: 60
        - type: 'TURN_ON'
        - type: 'WAIT'
          seconds: 60
        - type: 'TURN_OFF'
    - type: 'WAIT'
      seconds: 15
    - type: 'TARGET_TEMP'
      min: 30
      max: 35
    - type: 'REPEAT'
      times: 10
      steps:
        - type: 'WAIT'
          seconds: 15
        - type: 'TURN_ON'
        - type: 'WAIT'
          seconds: 15
        - type: 'TURN_OFF'
```

- Automatically controls power and temperature
- Defined by a dynamic schema including:
  - Power on/off
  - Wait
  - Set target temperature
  - Repeat
- Multiple programs can run simultaneously

PLC Service

- Interfaces with PLC safety system
- Observable for PLC status updates
  - Box temperature readings (separate from EBKeithley devices)
  - Box sensor statuses
  - Power supply interlock statuses
- Partially implemented by Pedja (thanks!)

Database Service

- Will record data from the setup in an SQL database
- Will observe all data-taking services, such as temperature, power, PLC, fan, etc.
- Not yet implemetned

Alert Service

- Allows services to send push notifications to individuals
- Utilizes the CERN Notifications Service
- Users/groups can choose how often they are notified and by what means (email, SMS, etc.)

Supervisor Service

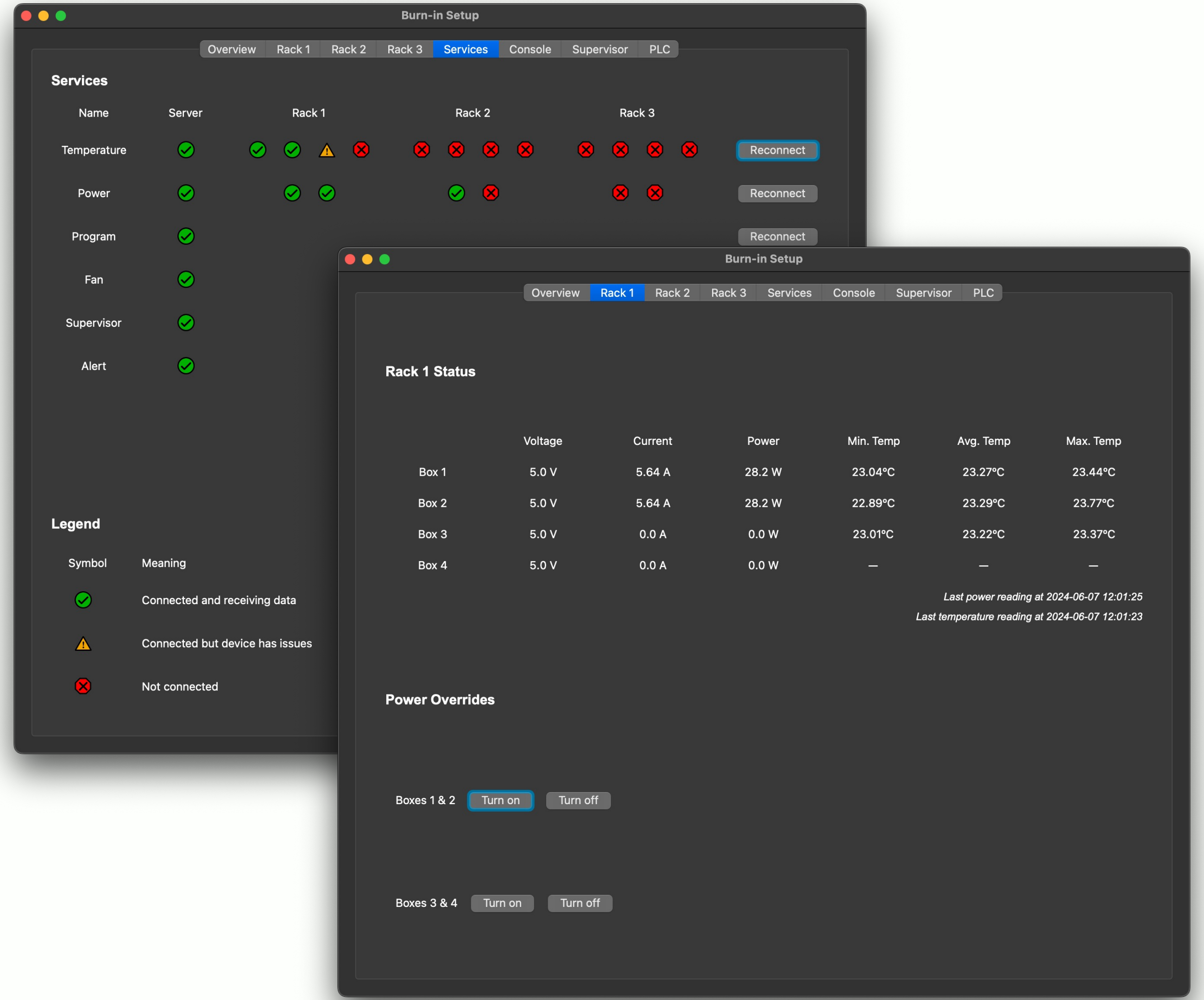
- Required for all other services to run
- Observes the temperature, power, and PLC services
- If it detects any anomaly (ex. temperature too high), will turn off all power supplies and alert users
- Polls temperature and power services to ensure they are still running properly

Orchestrator

- Used to start services in the correct order (based on their dependencies)
- Runs each service in a separate process
- Monitors services to detect if they stop running
- If a service is found to be stopped:
  - Kills the process
  - Restarts that service
  - Restarts all services which depend on that service

Graphical User Interface (GUI)

- Goals:
  - Provide necessary information at-a-glance
  - Allow emergency actions
  - Begin and monitor programs
- Connects to all services
- Built using PyQt 6 library



Services

Name	Server	Rack 1	Rack 2	Rack 3	
Temperature	✓	✓ ✓ ⚠ ✗	✗ ✗ ✗ ✗	✗ ✗ ✗ ✗	Reconnect
Power	✓	✓ ✓	✓ ✗	✗ ✗	Reconnect
Program	✓				Reconnect
Fan	✓				
Supervisor	✓				
Alert	✓				

Legend

Symbol	Meaning
✓	Connected and receiving data
⚠	Connected but device has issues
✗	Not connected

Rack 1 Status

	Voltage	Current	Power	Min. Temp	Avg. Temp	Max. Temp
Box 1	5.0 V	5.64 A	28.2 W	23.04°C	23.27°C	23.44°C
Box 2	5.0 V	5.64 A	28.2 W	22.89°C	23.29°C	23.77°C
Box 3	5.0 V	0.0 A	0.0 W	23.01°C	23.22°C	23.37°C
Box 4	5.0 V	0.0 A	0.0 W	—	—	—

Last power reading at 2024-06-07 12:01:25
Last temperature reading at 2024-06-07 12:01:23

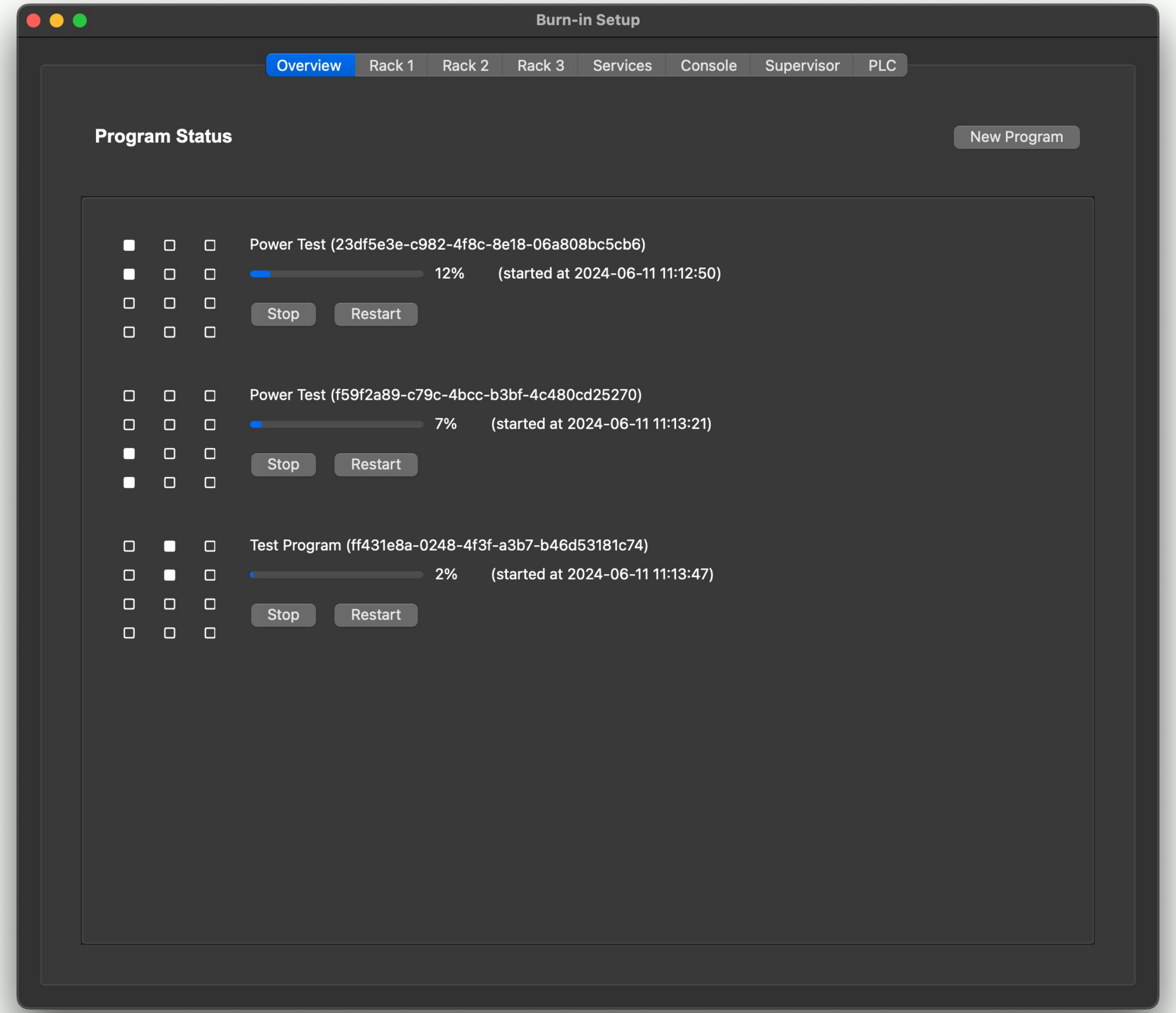
Power Overrides

Boxes 1 & 2 Turn on Turn off

Boxes 3 & 4 Turn on Turn off

Overview Tab

- Displays any active programs
  - Active boxes
  - Program name
  - Progress bar/percent
  - Stop/restart actions
- Ability to start new programs from a file



Program Status

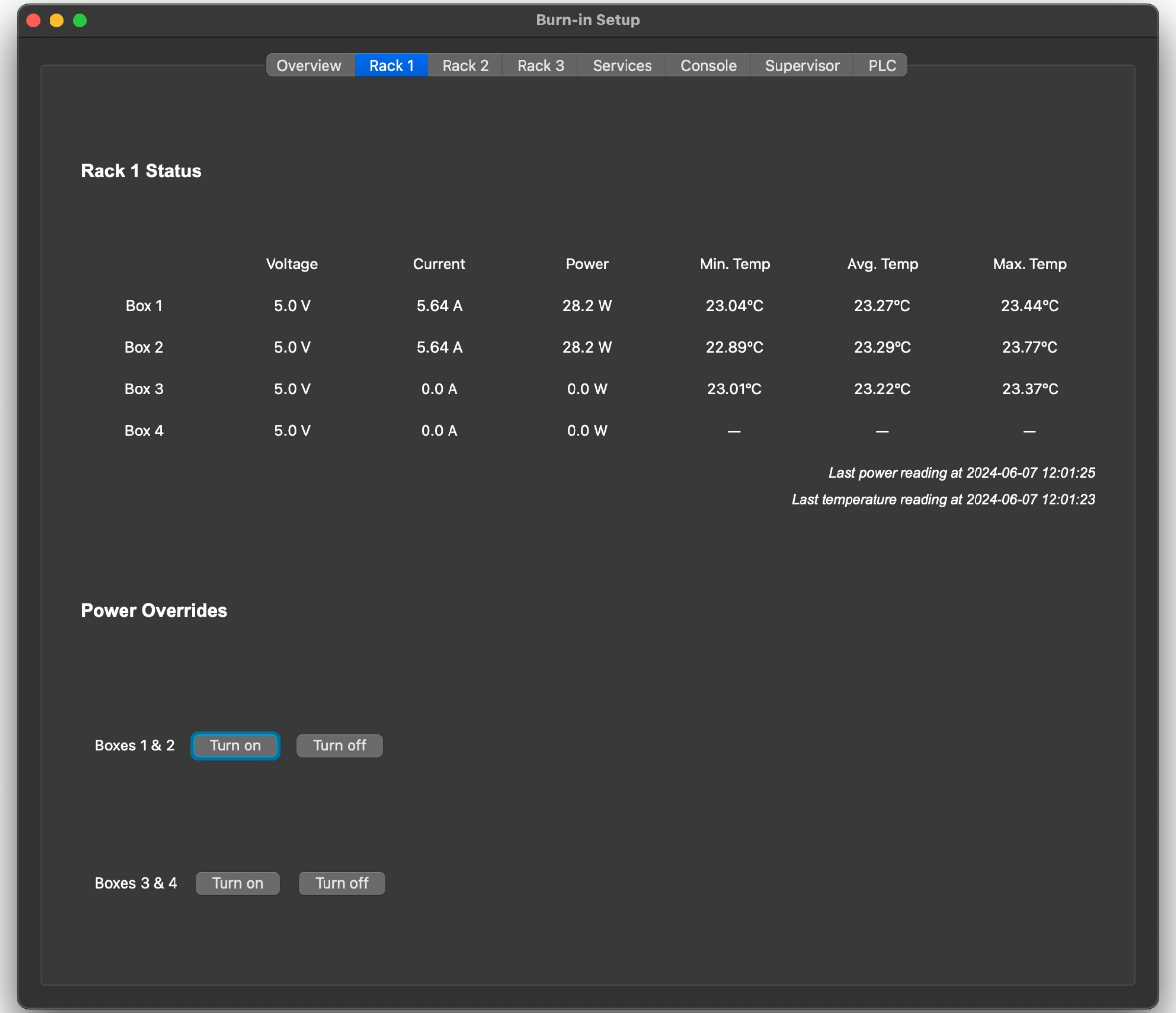
Power Test (23df5e3e-c982-4f8c-8e18-06a808bc5cb6)
12% (started at 2024-06-11 11:12:50)

Power Test (f59f2a89-c79c-4bcc-b3bf-4c480cd25270)
7% (started at 2024-06-11 11:13:21)

Test Program (ff431e8a-0248-4f3f-a3b7-b46d53181c74)
2% (started at 2024-06-11 11:13:47)

Rack Tabs

- Shows current temperature and power of each box
- Includes timestamps for when data was measured
- Provides overrides for turning power supplies on/off



Rack 1 Status

	Voltage	Current	Power	Min. Temp	Avg. Temp	Max. Temp
Box 1	5.0 V	5.64 A	28.2 W	23.04°C	23.27°C	23.44°C
Box 2	5.0 V	5.64 A	28.2 W	22.89°C	23.29°C	23.77°C
Box 3	5.0 V	0.0 A	0.0 W	23.01°C	23.22°C	23.37°C
Box 4	5.0 V	0.0 A	0.0 W	–	–	–

Last power reading at 2024-06-07 12:01:25
Last temperature reading at 2024-06-07 12:01:23

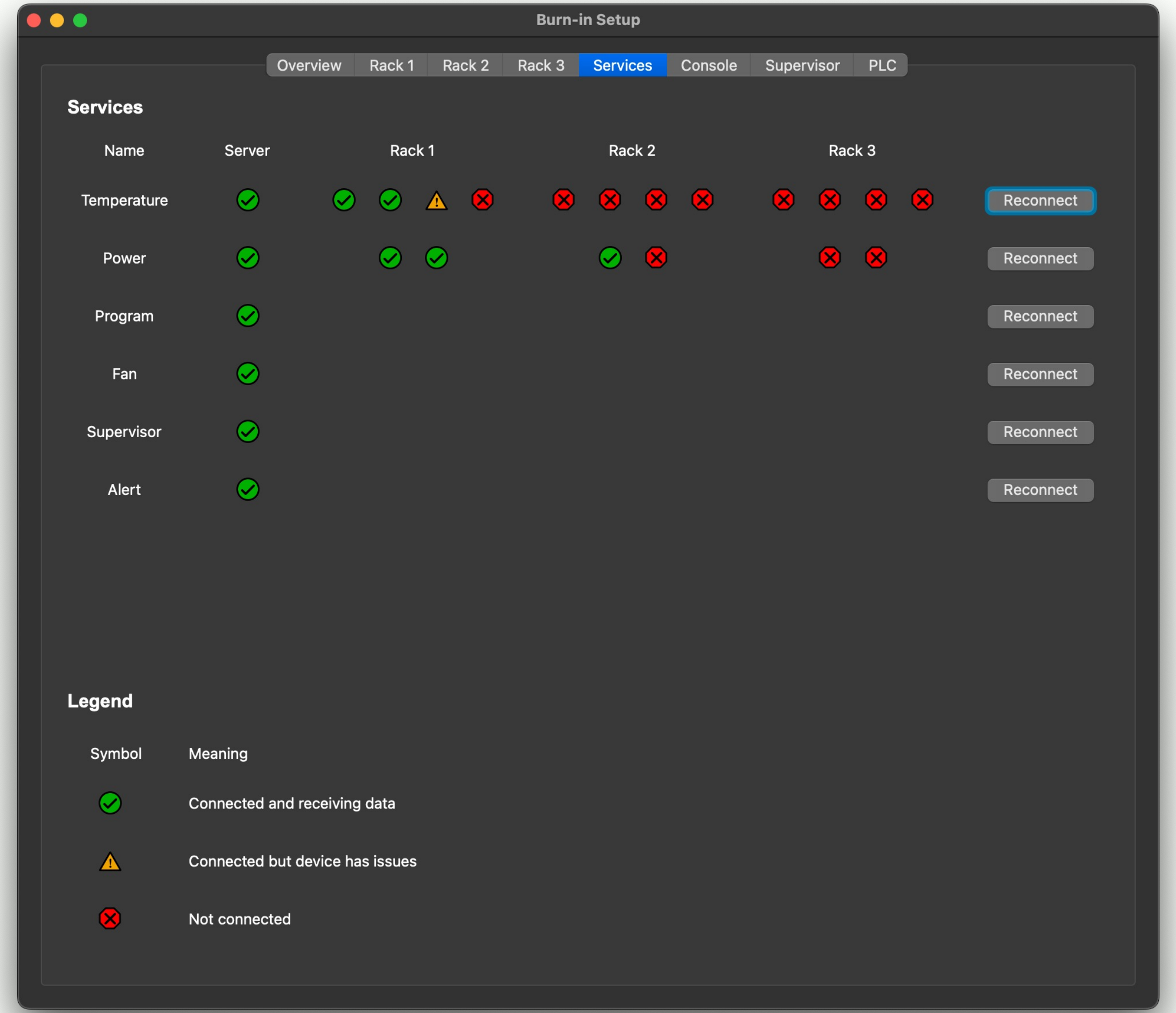
Power Overrides

Boxes 1 & 2 Turn on Turn off

Boxes 3 & 4 Turn on Turn off

Services Tab

- Displays whether services are connected to the GUI
- Can display device information
- Ability to reconnect, if a service is restarted after the GUI starts



Services

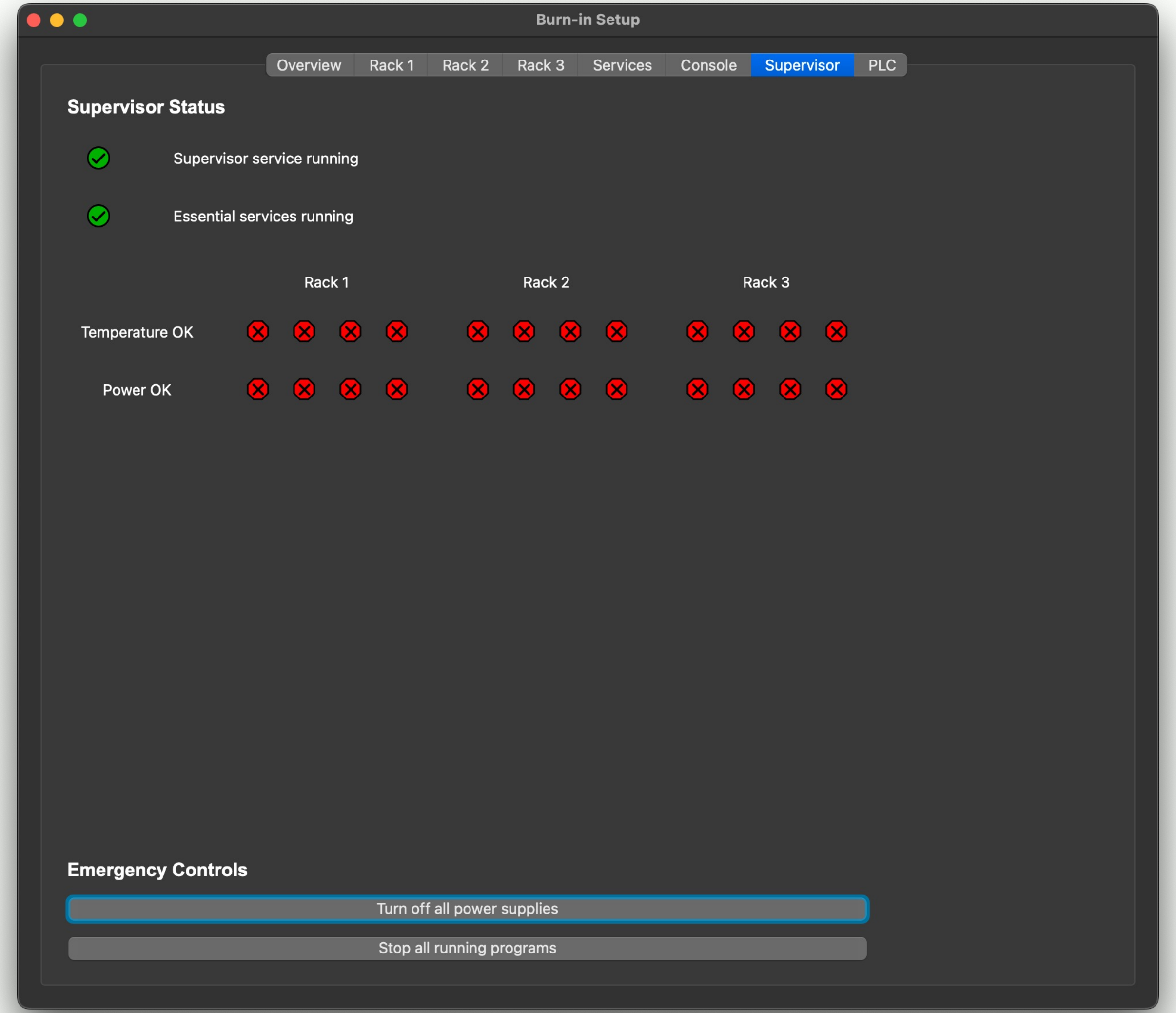
Name	Server	Rack 1				Rack 2				Rack 3				
Temperature	✓	✓	✓	⚠	✗	✗	✗	✗	✗	✗	✗	✗	✗	Reconnect
Power	✓		✓	✓			✓	✗			✗	✗		Reconnect
Program	✓													Reconnect
Fan	✓													Reconnect
Supervisor	✓													Reconnect
Alert	✓													Reconnect

Legend

Symbol	Meaning
✓	Connected and receiving data
⚠	Connected but device has issues
✗	Not connected

Supervisor Tab

- Shows status of the supervisor service
- Displays whether temperature and power measurements are within a safe range
- Provides emergency options
  - Turn off all power supplies
  - Stop all running programs



Burn-in Setup

Supervisor Status

✓ Supervisor service running

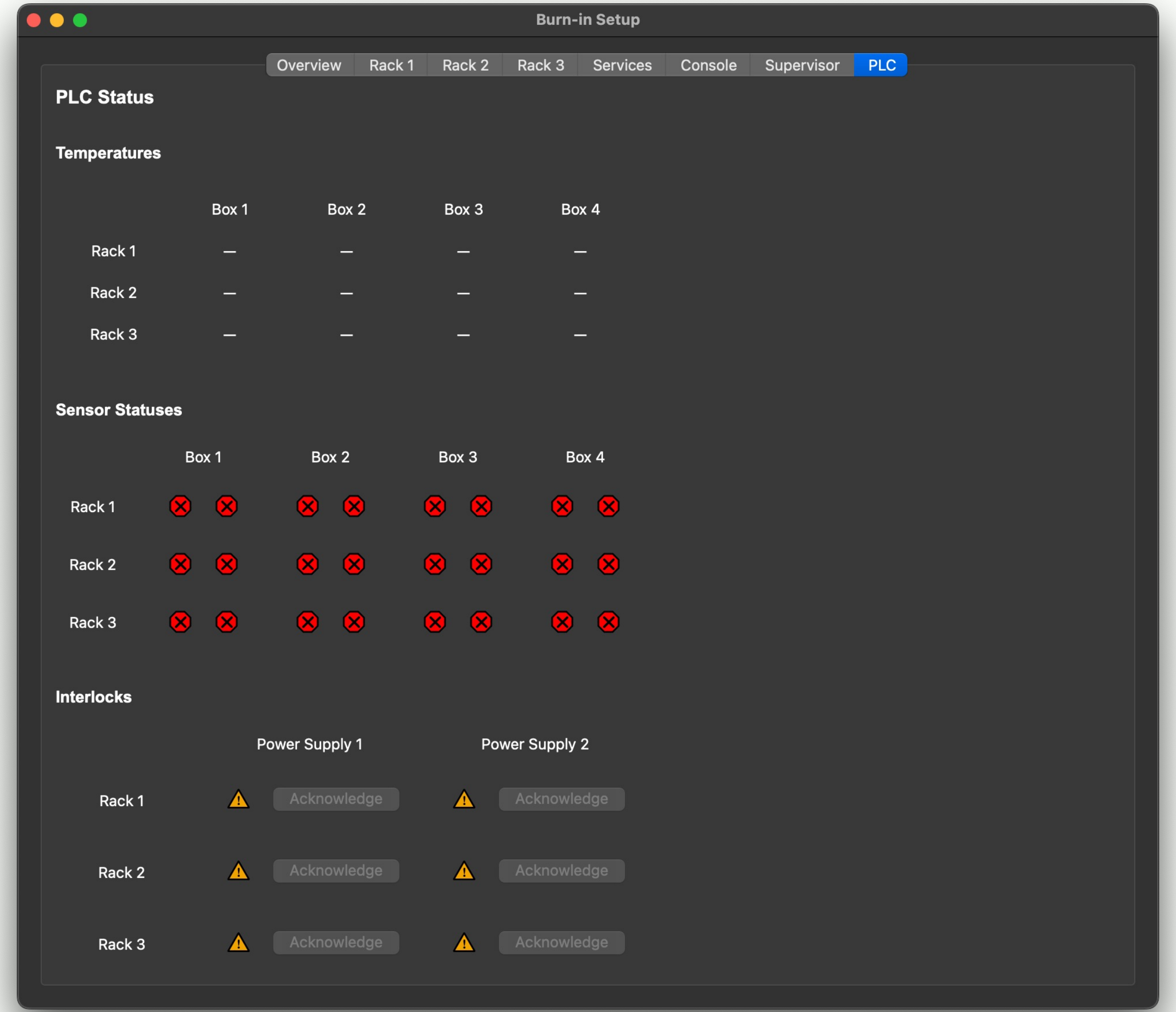
✓ Essential services running

	Rack 1				Rack 2				Rack 3			
Temperature OK	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Power OK	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Emergency Controls

PLC Tab

- Not yet connected to PLC service
- Will show PLC status
  - Box temperatures
  - Sensor statuses
  - Interlock statuses
- Ability to acknowledge interlocks



PLC Status

Temperatures

	Box 1	Box 2	Box 3	Box 4
Rack 1	–	–	–	–
Rack 2	–	–	–	–
Rack 3	–	–	–	–

Sensor Statuses

	Box 1		Box 2		Box 3		Box 4	
Rack 1	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
Rack 2	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
Rack 3	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗

Interlocks

	Power Supply 1		Power Supply 2	
Rack 1	⚠	Acknowledge	⚠	Acknowledge
Rack 2	⚠	Acknowledge	⚠	Acknowledge
Rack 3	⚠	Acknowledge	⚠	Acknowledge

Documentation

- README
    - Overview of project
    - Explanation of each service
- Services diagram
    - Relationships between services
- Code comments
    - Implementation-specific details
- Code guide
    - Detailed overview of code design decisions
- To-do list

Summary

Services implemented

- Temperature
- Power
- Fan (needs driver)
- Program
- Supervisor
- Alert

Services to be implemented

- PLC (mostly complete)
- Database

Summary

GUI functionality

- View current rack status
- Begin/monitor programs
- See service/device issues
- Take emergency action via supervisor
- PLC data (needs to be connected once service is done)

GUI functionality to be added

- PLC emergency controls
- Visualize temperatures/fan speeds in box
- Historical database viewer

Thank you!

Questions?