Mini-ULT NCAT Hack-a-thon

Problem Statement

Some biological materials, such as vaccines, require strict temperature control using a freezers. Customers storing these sensitive biological materials require secure audit logs of temperature over time to make sure that the freezer environment has maintained the appropriate temperature range and to track sample lifetime events. Additionally, manufacturers require the ability to collect usage and metrics for freezer failure and servicing.

As part of this NCAT Hack-a-thon Challenge, you will take an Mini-ULT Freezer, which currently outputs data via UART, and program it to collect and process metrics on the Communications Gateway.

Background

ULT Freezers are used to store samples or products that require precise temperature management, examples include:

- Vaccine storage at retail pharmacies
- Cold chain product storage
- Biological sample storage including biobanking and biorepository

Customers deployments may vary widely depending on the business use case from a single ULT Freezer in a pharmacy for storing vaccines and medication up to a warehouse with dozens or hundreds of ULT Freezers used for long term cold storage. In addition to designing ULT Freezers to meet customer expectations, manufacturers often include technology that collects usage and other metrics for predictive failure analysis and servicing the devices in the field.

Evaluation Criteria

Using the customer and manufacturer requirements and expectations presented in the Problem Statement, students will be scored based on a the implementation of the following features within the Mini-ULT Freezer and Communications Gateway environment.

Basic Mini-ULT and Communications Gateway Implementation

- Gateway app to view Mini-ULTs connected to the mesh/gateway and collect metrics:
  - View current state per connected Mini-ULT (Internal Temperature, Door State)
- Collect & process events from the Mini-ULT mesh devices on the Communications Gateway. Possible events that should be taken into account include:
  - Detect and alert if door is left open
  - Detect and alert if internal temperature falls outside of desired range
  - Detect unresponsive Mini-ULT node in the mesh
  - Mesh node offline
  - Power failure
  - Detect unresponsive Freezer Controller

Additional Features Beyond the Basic Implementation

Functional Features

- Use of LEDs for local alarm status or locating a specific freezer/sample.
  - Use LEDs to indicate alarm status if the freezer is not operating within configured parameters
  - Store fridge inventory in gateway application
  - Mobile app functionality could add identifying a freezer via BLE signal strength

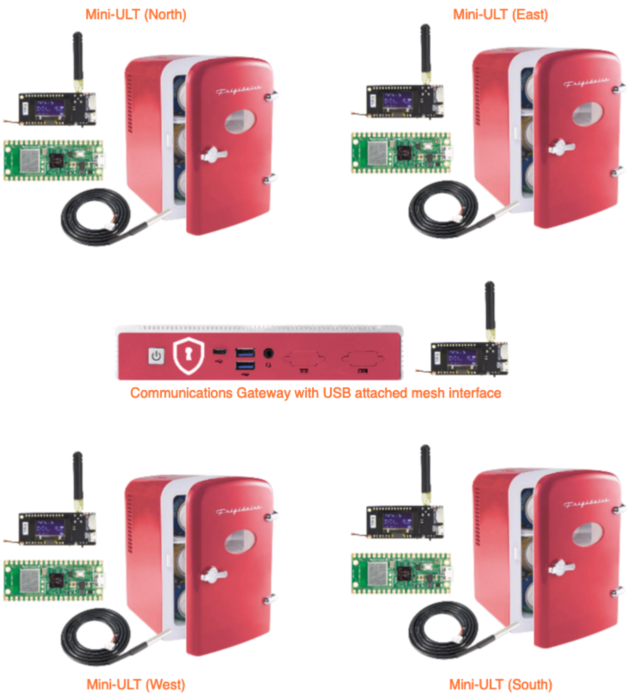
- Extend controller firmware to enable bi-directional communication to receive commands from gateway
    - Interface to "Locate" a specific freezer which triggers a command to change LED colors on the freezer
    - Interface to "Locate" a specific item which triggers a command to change LED colors on the freezer containing the item
    - Enable remote configuration of Mini-ULT parameters
        - Temperature set-point
        - Metric report interval
    - Enable remote control of Mini-ULT peripherals
        - LEDs

Cybersecurity Focused Features

- Threat Model of the Mini-ULT and Communications Gateway Environment

Hardware Environment

While there are a variety of architectures that can be implemented for ULT Freezers, for this Hack-a-thon the Mini-ULTs will be configured with a Communications Gateway that will simulate a basic product environment that requires additional features to meet customer and manufacturer needs.



Mini-ULT Freezer Ecosystem

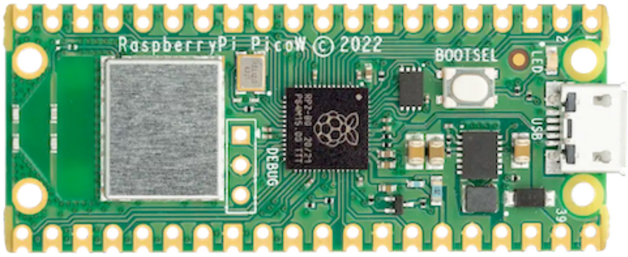
Hardware Breakdown

Full size ULT Freezers typically have four (4) separate controllers including a primary temperature controller, backup temperature controller, user interface controller, and communications controller.

For this Hack-a-thon the number of controllers has been simplified down to two controllers, including a fridge/freezer controller and communications controller.

Freezer Controller

- Platform: Microcontroller
- Hardware: Raspberry Pi Pico W
- Firmware: MicroPython or C (FreeRTOS / Arduino)
- Role: The role of the microcontroller will be to interface with sensors and actuators to control the Mini-ULT Freezer. Sensors and actuators available for this Hack-a-thon include:



Freezer Controller - Raspberry Pi Pico W

Sensors and Actuators for Freezer

- Temperature Sensor for measuring internal temp (DS18B20 Waterproof Temperature Sensor)
- Door State Sensor for detecting whether the door is open or closed (Simple Door Switch)
- Actuator for internal LED lights (NeoPixel RGB LED Strip)

Communications Controller

- Platform: Microcontroller
- Hardware: ESP32 LoRa Board
- Firmware: Meshtastic TLora (Written in C/C++ - FreeRTOS / Arduino)
- Role(s):
  - Run Meshtastic: An open source, off-grid, decentralized, mesh network built to run on affordable, low-power devices.
  - Interface with Fridge/Freezer Controller to provide mesh functionality for sending/receiving data. Example Configuration:
    - Serial Module connects Fridge/Freezer Controller (Pico W) via UART

Communications Controller - ESP32 LoRa Board

Communications Gateway

Communications Gateway

- Platform: General Purpose Computer (Intel NUC, etc.) or SBC (Raspberry PI 4, etc.)
- Hardware: Gateway and ESP32 LoRa Board, same as the Communications Controller.

- Role: Serve as a mesh gateway to run software that receives and processes data from Mini-ULT Freezers (nodes).

Software Environment

As part of this project you will be provided access to:

- Preconfigured and confirmed functional mesh environment on the Communications Controller.
- Minimal firmware on the Freezer Controller to measure temperature on a fixed interval and send output over UART to the Communications Controller.
- Communications Gateway with mesh interface attached and configured to receive simple temperature values from the Mini-ULTs.