Review the assignment due date

IoT Venture Pitch

ESE5180: IoT Wireless, Security, & Scaling

Team Name: The Circuiteers

Team Member Name	Email Address
Kevin Wang	wazhi@seas.upenn.edu
Shun Yao	shunyao@seas.upenn.edu
Zheyu Li	lizheyu@seas.upenn.edu
Megha Mistry	megham@seas.upenn.edu

GitHub Repository URL: https://github.com/ese5180/iot-venture-f25-the-circuiteers.git

Concept Development

The vending machine industry represents a $22-25 billion global market [2] with a critical need for smart health monitoring systems, as traditional maintenance approaches are bleeding operators dry with hidden costs that eliminate profits for 73% of independent vending operators [1]. Current maintenance expenses range from $100-500 annually per machine, with monthly checks costing $50-100 and service incidents adding $50-200 each time, creating a maintenance burden that can constitute up to 15% of total operational costs [1]. By using sensors to monitor component health in real-time—such as power draw, temperature, and dispensing motor function—operators can preempt failures before they happen, a practice that can reduce maintenance costs by up to 25% over time. [3] Take a mid-sized vending operator with a fleet of 100 machines, the total annual maintenance contract cost would be $60,000 ($600/machine x 100 machines). A 25% reduction means an annual saving of $15,000. A health monitoring system prevents costly emergency repairs, optimizes service routes to eliminate unnecessary trips, and cuts product waste from overstocking or machine malfunctions. By transforming vending machines from static boxes into efficient, reliable, and more profitable assets, a smart health monitoring system provides a clear and rapid return on investment.

Sources:

1. https://dfyvending.com/vending-machine-maintenance-costs
2. https://www.nextmsc.com/report/vending-machine-market
3. https://businessplan-templates.com/blogs/startup-costs/vending-machine
4. https://www.news.market.us/vending-machine-statistics/

Product Function

The product is a smart health monitoring system for vending machines that uses sensors to track critical components like temperature and motor vibration in real-time. It wirelessly transmits health data from distributed vending machines to a central processing system, enabling operators to monitor hundreds of

machines simultaneously. The system provides predictive maintenance capabilities by identifying potential failures before they occur, helping vending operators reduce maintenance costs by up to 25% and prevent costly emergency repairs.

Target Market & Demographics

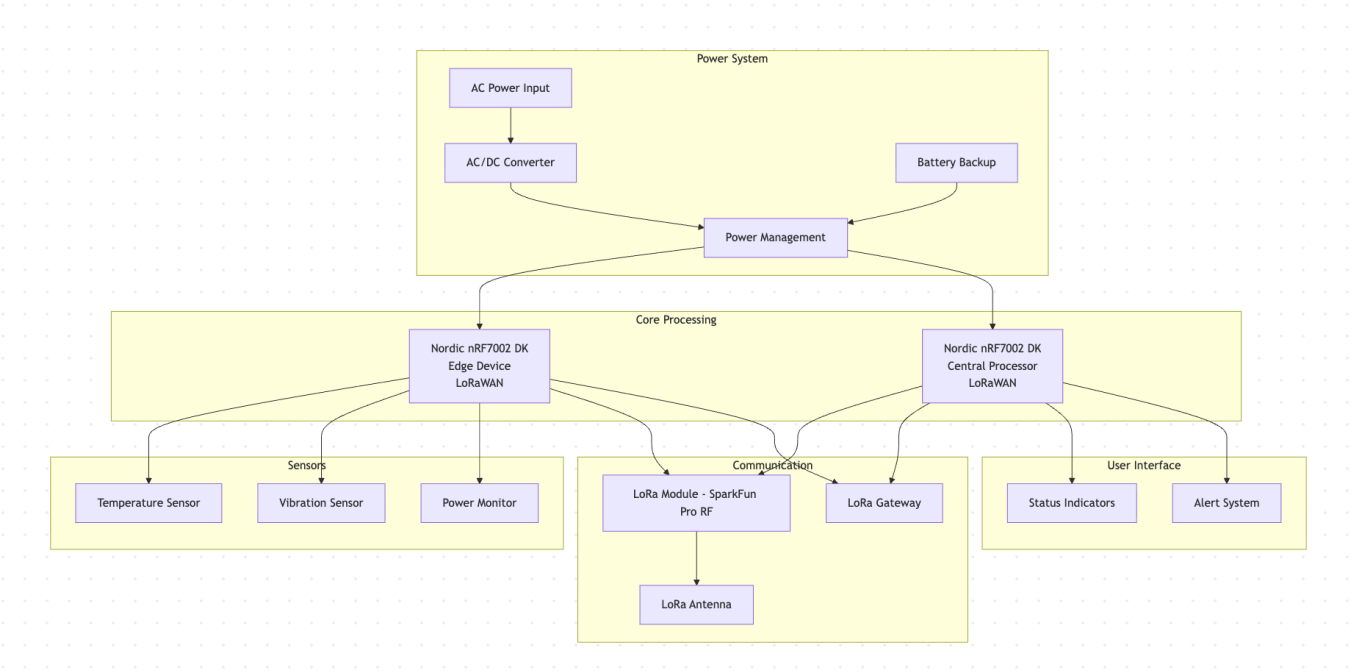
- Primary Users: Vending machine operators, fleet managers, and maintenance technicians who need real-time visibility into machine health across distributed locations.
- Primary Customers: Commercial vending companies, facilities management companies, and large-scale vending operators managing 50+ machines.
- Geographic Deployment: Initially targeting the United States market, with expansion planned for urban areas in Canada and Europe where vending machine density is highest.
- Market Size: The global vending machine market size in 2024 was USD 23065.3 million, and is estimated to reach USD 25879.3 million in 2025 and is projected to reach USD 34959.4 million by 2030 with a CAGR of 6.20%. According to NMSC, in term of volume the market size was 12 million units in 2024, and is estimated to reach 15 million units in 2025 and is projected to reach 23 million units by 2030 with a CAGR of 9.3%. Source: https://www.nextmsc.com/report/vending-machine-market.
- Competitors: While asset monitoring systems (Augury) and smart (with capabilities to track sales, suggest products to purchase, do inventory monitring etc.) vending machines exist (BetterHealth Vending, SMRT1 Technologies, Velocity Smart), there appears to be a gap in the area of health monitoring systems specifically for vending machines.

Stakeholders

Jeff Dreyer serves as our key stakeholder and industry advisor for this vending machine health monitoring project. As Senior Director of Product Management and Enablement at Augmentir, Jeff brings over 10 years of experience in industrial IoT solutions and product development. His background spans from software engineering at ThingWorx (PTC) to his current role leading product enablement at Augmentir, where he focuses on connected worker solutions and industrial asset management. Jeff's expertise in IoT platforms, product strategy, and industrial automation makes him an invaluable advisor for developing our smart health monitoring system for vending machines. His insights will help ensure our solution aligns with industry best practices and market needs.

System-Level Diagrams

Device Block Diagram

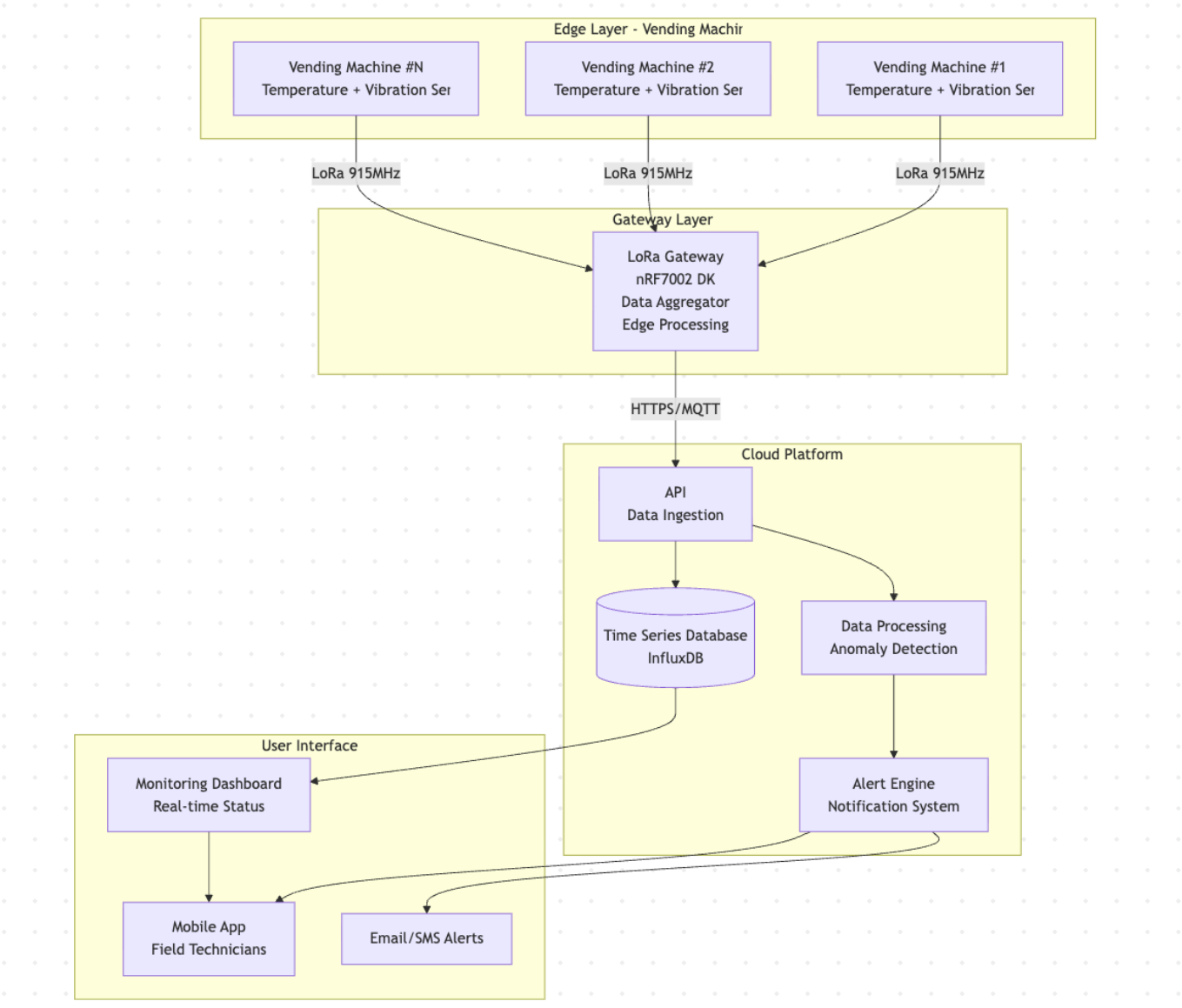


Source Code:

```
graph TB
    subgraph "Power System"
        AC[AC Power Input] --> PSU[AC/DC Converter]
        PSU --> PMIC[Power Management]
        BAT[Battery Backup] --> PMIC
    end

    subgraph "Core Processing"
        EDGE[Nordic nRF7002 DK
Edge Device
LoRaWAN]
        CENTRAL[Nordic nRF7002 DK
Central Processor
LoRaWAN]
    end

    subgraph "Sensors"
        TEMP[Temperature Sensor]
        VIB[Vibration Sensor]
        POWER[Power Monitor]
    end

    subgraph "User Interface"
        LED[Status Indicators]
        BUZZER[Alert System]
    end

    subgraph "Communication"
        LORA[LoRa Module - SparkFun Pro RF]
        GATE[LoRa Gateway]
        ANT[LoRa Antenna]
```

```
    end

    PMIC --> EDGE
    PMIC --> CENTRAL
    EDGE --> TEMP
    EDGE --> VIB
    EDGE --> POWER
    EDGE --> LORA
    CENTRAL --> LORA
    LORA --> ANT
    CENTRAL --> GATE
    EDGE --> GATE
    CENTRAL --> LED
    CENTRAL --> BUZZER
```

Communication Architecture Diagram



Source Code:

```
graph TB
    subgraph "Edge Layer - Vending Machines"
        VM1[Vending Machine #1
Temperature + Vibration Sensors + more]
        VM2[Vending Machine #2
Temperature + Vibration Sensors + more]
        VM3[Vending Machine #N
Temperature + Vibration Sensors + more]
    end

    subgraph "Gateway Layer"
        GW[LoRa Gateway
nRF7002 DK
Data Aggregator
Edge Processing]
    end

    subgraph "Cloud Platform"
        API[API
Data Ingestion]
        DB[(Time Series Database
InfluxDB)]
        PROC[Data Processing
Anomaly Detection]
        ALERT[Alert Engine
Notification System]
    end

    subgraph "User Interface"
        DASH[Monitoring Dashboard
Real-time Status]
        MOBILE[Mobile App
Field Technicians]
        EMAIL[Email/SMS Alerts]
    end

    VM1 -->|LoRa 915MHz| GW
    VM2 -->|LoRa 915MHz| GW
    VM3 -->|LoRa 915MHz| GW

    GW -->|HTTPS/MQTT| API
    API --> DB
    API --> PROC
    PROC --> ALERT
    DB --> DASH
    ALERT --> EMAIL
    ALERT --> MOBILE
    DASH --> MOBILE
```

Security Requirements Specification

The vending machine health monitoring system requires robust security measures to protect operational data, prevent unauthorized access to machine controls, and ensure the integrity of maintenance predictions. Security is critical as compromised systems could lead to false maintenance alerts, data theft, or disruption of vending operations.

SEC 01 – All LoRaWAN communications shall use standard LoRaWAN 1.0.x security with unique device keys to prevent unauthorized devices from joining the network.

SEC 02 – The cloud API shall require authentication tokens for all data submissions to prevent unauthorized data injection.

SEC 03 – Firmware updates shall include checksums to verify integrity and prevent corrupted updates from bricking devices.

Hardware Requirements Specification

The hardware platform consists of distributed sensor nodes deployed in vending machines, communicating via LoRa to a central gateway. The system must operate reliably in commercial environments with varying temperatures and electrical noise while maintaining low power consumption for battery-backed operation.

HRS 01 – The system shall use Nordic nRF7002 DK as the primary microcontroller platform for both edge devices and central gateway, operating at 3.3V with integrated Wi-Fi/Bluetooth capabilities.

HRS 02 – Temperature monitoring shall use I2C-based digital temperature sensors with ±5°C accuracy across -10°C to +110°C range, sampling at 1Hz minimum.

HRS 03 – Vibration detection shall utilize a 3-axis MEMS accelerometer with 16-bit resolution to monitor motor health, sampling at 100Hz minimum.

HRS 04 – The SparkFun Pro RF LoRa module shall provide 915MHz communication with minimum -148dBm sensitivity and +20dBm transmit power for reliable in-building coverage.

Software Requirements Specification

The software system manages distributed sensor data collection, wireless transmission, edge processing, and predictive maintenance algorithms. Built on Zephyr RTOS, it ensures real-time performance while maintaining power efficiency for battery-operated edge nodes.

SRS 01 – The edge device firmware shall collect temperature, vibration, and power consumption data every 60 seconds (±5 seconds) and transmit via LoRaWAN to the central gateway.

SRS 02 – The system shall implement edge anomaly detection using a sliding window algorithm to identify temperature deviations or vibration patterns.

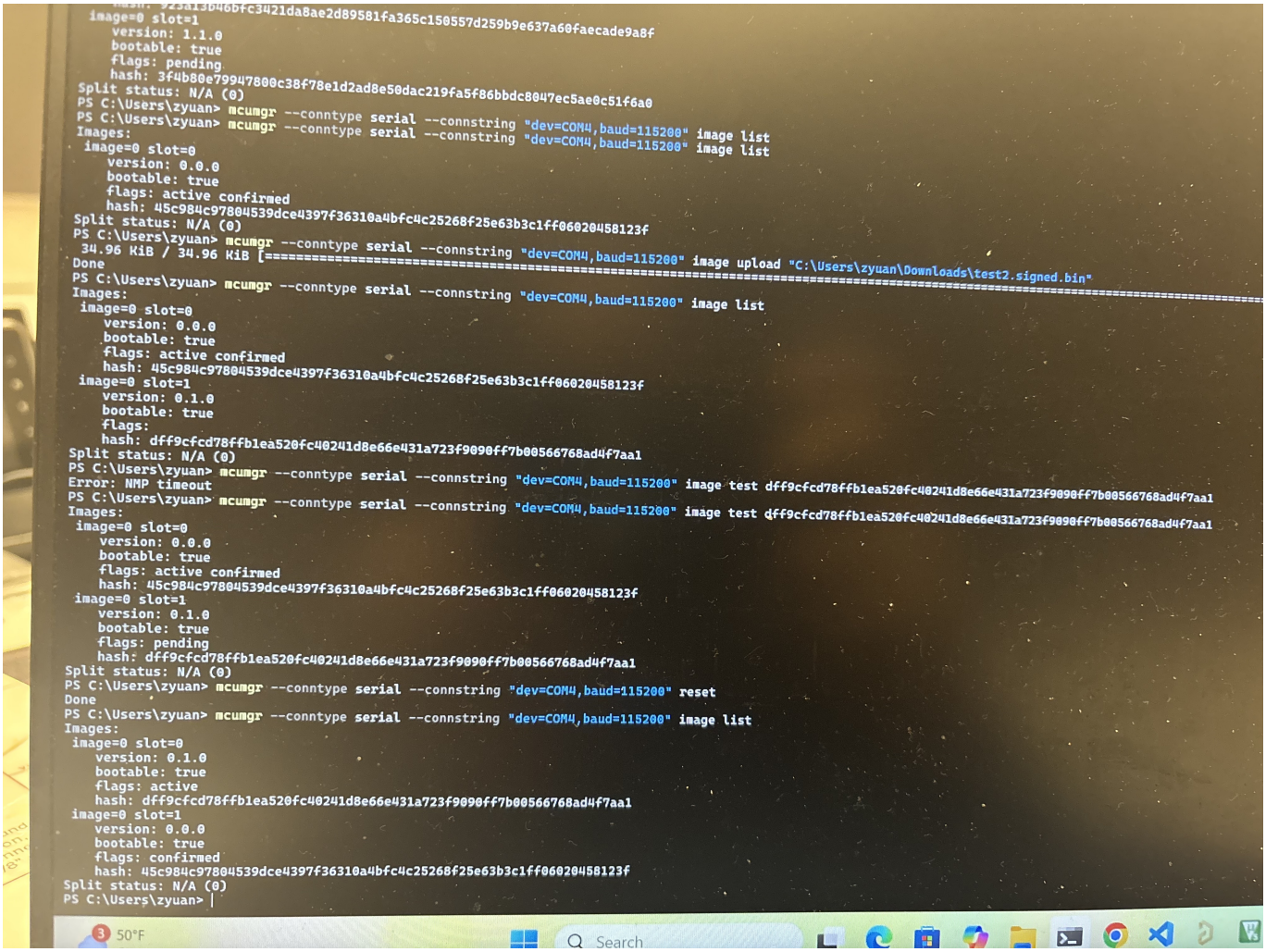
SRS 03 – The gateway shall aggregate data from the vending machine and forward processed metrics to the cloud platform via MQTT/HTTPS every 5 minutes.

SRS 04 – The software shall maintain a local data buffer capable of storing 24 hours of sensor readings in case of communication failure, with automatic retransmission upon connection restoration.

Secure Firmware Updates

Extra Credit (EC1): Implement serial recovery with signed images over UART through the bootloader as a failsafe backup

Screenshot of the successful implementation of serial recovery with signed images over UART:



For the code implementation, refer to the `boot_with_keys/` directory.

Bootloading Process Overview

Our VendGuard system implements a secure dual-slot bootloading architecture using MCUboot with cryptographic signature verification and LoRaWAN-based Firmware Update Over The Air (FUOTA) capabilities.

Bootloading Architecture Diagram

```
graph TB
    subgraph "STM32WL55JC Flash Memory (256KB)"
        BOOT["MCUboot Bootloader
36KB @ 0x0000"]
        SLOT0["Primary App Slot (slot0)
102KB @ 0x9000"]
        SLOT1["Secondary App Slot (slot1)
102KB @ 0x22800"]
        STORAGE["Storage Partition
8KB @ 0x3C000"]
```

```
        SCRATCH["Scratch Area
8KB @ 0x3E000"]
    end

    subgraph "Boot Process Flow"
        POWER["Power On/Reset"] --> BOOT
        BOOT --> VERIFY["Verify Slot0
Signature"]
        VERIFY -->|Valid| RUN["Run Application
from Slot0"]
        VERIFY -->|Invalid| RECOVERY["Serial Recovery
Mode (UART)"]

        UPDATE["FUOTA Update
Available"] --> DOWNLOAD["Download to Slot1
via LoRaWAN"]
        DOWNLOAD --> VALIDATE["Validate New
Image Signature"]
        VALIDATE -->|Valid| SWAP["Mark for Swap
& Reboot"]
        VALIDATE -->|Invalid| REJECT["Reject Update"]
        SWAP --> BOOT
    end

    subgraph "FUOTA Process"
        LORAWAN["LoRaWAN Network"] --> FRAG["Fragmented
Data Transport"]
        FRAG --> REASSEMBLE["Reassemble
Firmware Image"]
        REASSEMBLE --> SLOT1
    end

    LORAWAN -.-> UPDATE
    RUN --> UPDATE
```

Bootloader and Application Sizes

MCUboot Bootloader:

- **Size**: 36,284 bytes (35.4 KB)
- **Allocated Space**: 36 KB (36,864 bytes)
- **Location**: Flash base address 0x08000000

Application Code:

- **Size**: 35,660 bytes (34.8 KB) for basic LED blink application
- **Allocated Space**: 102 KB (104,448 bytes) per slot
- **Utilization**: 34.1% of allocated space (allows room for sensor integration and LoRaWAN stack)
- **Location**: Slot0 at 0x08009000, Slot1 at 0x08022800

Firmware Update Process

Download Mechanism

Our FUOTA implementation uses **LoRaWAN (915 MHz ISM band)** as the wireless communication protocol, specifically LoRaWAN 1.0.4 with OTAA (Over-the-Air Activation) for secure device authentication. The system employs the Fragmented Data Transport service to handle large firmware images efficiently, and switches to Class C during multicast sessions for efficient bulk data transfer to multiple devices simultaneously.

Why LoRaWAN for FUOTA?

We chose LoRaWAN over other wireless methods because:

1. **Long Range**: Up to 2km coverage in urban environments, ideal for distributed vending machines
2. **Low Power**: Battery-operated edge devices can receive updates without frequent charging
3. **Multicast Capability**: Single transmission can update multiple devices simultaneously
4. **Standardized FUOTA**: Built-in LoRaWAN services handle fragmentation, reassembly, and scheduling
5. **Existing Infrastructure**: Leverages the same communication channel used for sensor data

Firmware Storage Location

Downloaded firmware images are stored in:

- **Primary Location**: Slot1 (secondary application slot) at flash address 0x08022800
- **Size**: 102 KB maximum per image
- **Temporary Storage**: 8 KB scratch area at 0x0003E000 for image swapping operations

Failure Handling Features

Our system implements multiple layers of protection against firmware update failures:

1. Cryptographic Signature Verification

- **Algorithm**: ECDSA P-256 with SHA-256
- **Key Management**: Private key stored securely, public key embedded in bootloader
- **Validation Points**:
    - Before storing downloaded image in Slot1
    - Before swapping images during boot process
    - **Protection**: Prevents installation of unsigned or tampered firmware

2. Image Integrity Checks

- **CRC Validation**: Fragmented transport includes CRC checks for each fragment
- **Hash Verification**: Complete image hash validation after reassembly
- **Size Validation**: Ensures image fits within allocated slot space
- **Protection**: Detects corrupted downloads or transmission errors

3. Dual-Slot Failsafe Architecture

- **Swap Mechanism**: Uses scratch area for atomic image swapping
- **Rollback Protection**: Original firmware remains in Slot0 until new image is verified

- **Boot Validation**: MCUboot validates Slot0 signature on every boot
- **Protection**: System can always boot from known-good firmware

4. Serial Recovery Mode

- **Activation**: Automatic entry when no valid image found in Slot0
- **Interface**: UART-based recovery protocol
- **LED Indication**: Blue LED indicates recovery mode active
- **Capabilities**:
  - Emergency firmware upload via UART
  - Image validation and installation
  - **Protection**: Provides recovery path when both slots are corrupted

5. Upgrade Confirmation

- **Test Boot**: New firmware runs in "test" mode initially
- **Confirmation Required**: Application must confirm successful operation
- **Automatic Rollback**: Reverts to previous firmware if confirmation not received
- **Timeout**: Configurable timeout period for confirmation
- **Protection**: Prevents permanent installation of faulty code with valid signatures

Security Features

- **Encrypted Communication**: LoRaWAN AES-128 encryption for all transmissions
- **Device Authentication**: OTAA with unique device keys prevents unauthorized updates
- **Replay Protection**: LoRaWAN frame counters prevent replay attacks
- **Code Signing**: ECDSA P-256 signatures ensure firmware authenticity and integrity

CI/CD Pipeline

Continuous Integration (CI) Pipeline

The project uses GitHub Actions for continuous integration to automatically build and verify firmware for all target platforms.

Location: `.github/workflows/ci.yml`

What it does:

- Automatically builds firmware for all applications on every push and pull request
- Builds for target board: `nucleo_wl55jc` (STM32WL55xx)
- Applications built:
  - `apps/` - Main LoRaWAN application for edge nodes
  - `fuota/` - Firmware update over the air application
  - `boot_with_keys/` - Secure bootloader with keys
  - `scratch_test/` - Scratch test application
  - `uart_boot/` - UART bootloader application
- Verifies build artifacts are generated correctly
- Uploads build artifacts for download (retained for 7 days)

View CI Status:

- Go to the "Actions" tab in GitHub to see build status
- Green checkmark = all builds successful
- Red X = one or more builds failed

Benefits:

- Catches compilation errors before merging code
- Ensures code builds on clean environment
- Provides downloadable firmware binaries
- Documents build process for team members

Git Hooks

Git hooks are configured to maintain code quality and catch issues early in the development process.

Location: `hooks/` directory

Available Hooks

Pre-commit Hook

Runs automatically before each commit to perform quick checks:

- Detects merge conflict markers (blocks commit)
- Warns about TODO/FIXME comments
- Warns about trailing whitespace
- Warns about large files (>10MB)
- Basic syntax validation for C files
- **Runs Mock Unit Tests** from `sensors_test/tests_unit_mock/` (if available)
  - Automatically builds tests if needed
  - Blocks commit if tests fail
  - Requires `cmake` and `ctest` to be installed

Pre-push Hook

Runs automatically before pushing to remote repository:

- Verifies that the main application builds successfully
- Prevents pushing code that doesn't compile
- Uses 5-minute timeout to prevent hanging
- **Runs Mock Unit Tests** from `sensors_test/tests_unit_mock/` (if available)
  - Ensures all tests pass before pushing
  - Requires `cmake` and `ctest` to be installed

Installation

Quick Install:

```
chmod +x hooks/install.sh
./hooks/install.sh
```

Manual Install:

```
cp hooks/pre-commit .git/hooks/pre-commit
cp hooks/pre-push .git/hooks/pre-push
chmod +x .git/hooks/pre-commit
chmod +x .git/hooks/pre-push
```

Usage

Hooks run automatically:

- `git commit` → triggers pre-commit hook
- `git push` → triggers pre-push hook

To skip hooks (not recommended):

```
git commit --no-verify
git push --no-verify
```

Requirements

- **Pre-commit:**
  - No special requirements for basic checks
  - `cmake` and `ctest` required for unit tests (optional, tests skipped if not available)
- **Pre-push:**
  - Requires Zephyr environment (`source zephyr/zephyr-env.sh`) and `west` tool for build verification
  - `cmake` and `ctest` required for unit tests (optional, tests skipped if not available)

For detailed documentation, see hooks/README.md.

Concept Refinement

The vending machine industry represents a $22-25 billion global market with a critical need for smart health monitoring systems, as traditional maintenance approaches are bleeding operators dry with hidden costs that eliminate profits for 73% of independent vending operators. Current maintenance expenses range from $100-500 annually per machine, with monthly checks costing $50-100 and service incidents adding $50-200 each time, creating a maintenance burden that constitutes up to 15% of total operational costs. VendGuard is a plug-and-play IoT solution that transforms maintenance from reactive to predictive, using sensors to monitor critical vending machine components in real-time—including power draw, temperature, dispensing motor function, and cooling system performance. By identifying potential failures before they occur, VendGuard reduces maintenance costs by up to 25%, prevents emergency repairs, optimizes service routes, and extends

machine lifespan. For a mid-sized vending operator with 100 machines, VendGuard can provide annual maintenance savings of $15,000 ($600/machine × 100 machines × 25% reduction), while simultaneously increasing machine uptime, reducing product spoilage, and improving customer satisfaction.

Product Function

VendGuard is a smart health monitoring system for vending machines that leverages LoRaWAN connectivity and intelligent sensors to track critical components. The system has three main components:

1. Edge Device: A compact, battery-backed sensor package that monitors temperature, vibration and dispensing mechanism health. The device attaches non-invasively to existing vending machines and wirelessly transmits data via LoRaWAN.
2. Gateway: A centralized hub that collects data from multiple VendGuard devices within range (up to 2km in urban environments) and forwards the aggregated data to the cloud platform.
3. Cloud Platform: A web-based dashboard that provides real-time machine status, predictive maintenance alerts, service scheduling tools, and historical performance analytics.

The system identifies failure patterns before they cause downtime by analyzing sensor data using machine learning algorithms. It generates maintenance alerts with specific component recommendations, optimizes technician routing, and provides comprehensive fleet health reporting.

Target Market & Demographics

• **Primary Users**: Vending machine operators, fleet managers, and maintenance technicians who need real-time visibility into machine health across distributed locations. • **Primary Customers**: Commercial vending companies, facilities management firms, and vending operators managing 50+ machines who currently spend 5% of operational budget on emergency repairs. • **Addressable Market**: o Stage 1 (United States): Serviceable addressable market representing 30% of the 8 million machines owned by small operators who would benefit from predictive maintenance. o Stage 2 (Canada + Europe): With expansion into urban areas of Canada and Europe, our addressable market grows to 2.1M machines where vending density is highest. • **Market Size**: The global vending machine market was $23.1 billion in 2024 and is projected to reach $34.9 billion by 2030 (CAGR of 6.2%). In volume, the market is expected to grow from 12 million units in 2024 to 23 million units by 2030 (CAGR of 9.3%). • **Customer Profile**: Target customers are 75% of vending operators managing 50+ machines who still use manual maintenance schedules. • **Competitors**: While asset monitoring systems (Augury) and smart (with capabilities to track sales, suggest products to purchase, do inventory monitring etc.) vending machines exist (BetterHealth Vending, SMRT1 Technologies, Velocity Smart), there appears to be a gap in the area of health monitoring systems specifically for vending machines.

Stakeholders

Jeff Dreyer serves as our key stakeholder and industry advisor for this vending machine health monitoring project. As Senior Director of Product Management and Enablement at Augmentir, Jeff brings over 10 years of experience in industrial IoT solutions and product development. His background spans from software engineering at ThingWorx (PTC) to his current role leading product enablement at Augmentir, where he focuses on connected worker solutions and industrial asset management. Jeff's expertise in IoT platforms, product strategy, and industrial automation makes him an invaluable advisor for developing our smart health monitoring system for vending machines. His insights will help ensure our solution aligns with industry best practices and market needs.

System-Level Diagrams

Device Power Block Diagram

Source Code:

```
graph TB
    subgraph G1["Gateway Power Supply"]
        OUTLET["Wall Outlet
(AC 120V or 220V)"]
        GW["LoRa Gateway"]
        OUTLET --> GW
    end

    subgraph G2["Edge Node Power Supply
Vending Machine"]
        BATT["Battery Pack
(Li-ion 3.6V/3.7V)"]
        NUCLEO["Nucleo WL55jc1 Board"]
        SENSORS["Sensors
(Temperature, Vibration)"]
        BATT --> NUCLEO --> SENSORS
    end

    G1 -.-> G2
```

Communication Architecture Diagram

Source Code:

```
graph TB
    subgraph "Edge Layer Machines"
        VM1[Vending Machine #1
Nucleo WL55jc1
 + Temperature & Vibration Sensors + more]
        VM2[Vending Machine #2
Nucleo WL55jc1
  + Temperature & Vibration Sensors + more]
        VM3[Vending Machine #N
Nucleo WL55jc1
  + Temperature & Vibration Sensors + more]
    end

    subgraph "Gateway Layer"
        GW[LoRa Gateway
Multicast Group]
    end

    subgraph "AWS Cloud Service"
```

```
        API[API
Data Ingestion]
        DB[Database]
        PROC[Data Processing
Anomaly Detection]
        ALERT[Alert Engine
Notification System]
    end

    subgraph "User Interface"
        MOBILE[Mobile App
Field Technicians]
        EMAIL[Email/SMS Alerts]
    end

    VM1 <-->|LoRa 915MHz| GW
    VM2 <-->|LoRa 915MHz| GW
    VM3 <-->|LoRa 915MHz| GW

    GW <-->|WiFi| API
    API --> DB
    API --> PROC
    PROC --> ALERT
    ALERT --> EMAIL
    ALERT --> MOBILE
```

Security Requirements Specification

The vending machine health monitoring system requires robust security measures to protect operational data, prevent unauthorized access to machine controls, and ensure the integrity of maintenance predictions. Security is critical as compromised systems could lead to false maintenance alerts, data theft, or disruption of vending operations.

SEC 01 – All LoRaWAN communications shall use standard LoRaWAN 1.0.4 security with unique device keys and OTAA (Over-the-Air Activation) to prevent unauthorized devices from joining the network.

SEC 02 – The cloud API shall require authentication tokens for all data submissions to prevent unauthorized data injection.

SEC 03 – Firmware updates shall include checksums to verify integrity and prevent corrupted updates from bricking devices.

SEC 03 – The system shall implement role-based access controls to restrict dashboard and API access based on user permissions.

Hardware Requirements Specification

The hardware platform consists of distributed sensor nodes deployed in vending machines, communicating via LoRa to a central gateway. The system must operate reliably in commercial environments with varying temperatures and electrical noise while maintaining low power consumption for battery-backed operation.

HRS 01 – The edge device shall use the STM32WL55JC1 as the primary microcontroller platform, leveraging its integrated SX126x LoRa radio and ultra-low-power features for extended battery life.

HRS 02 – Temperature monitoring shall use the TMP117 I2C-based digital temperature sensor with ±0.1°C accuracy across -10°C to +85°C range, sampling at 1Hz minimum.

HRS 03 – Vibration detection shall utilize the ADXL345 3-axis MEMS accelerometer with 16-bit resolution connected via SPI interface to monitor motor health, sampling at 100Hz during active periods.

HRS 04 – The system shall utilize the Browan MiniHub Pro as the LoRaWAN gateway, supporting up to 100 device connections with 8-channel capability and backhaul connectivity via Wi-Fi, Ethernet or LTE.

HRS 05 – The edge device shall be powered by a 3000mAh Li-Po battery, providing a minimum 6-month operational life without recharging.

Software Requirements Specification

The software system manages distributed sensor data collection, wireless transmission, edge processing, and predictive maintenance algorithms.

SRS 01 – The edge device firmware shall collect temperature, vibration, and power consumption data every 60 seconds (±5 seconds) and transmit via LoRaWAN to the central gateway.

SRS 02 – The system shall implement edge anomaly detection using a sliding window algorithm to identify temperature deviations or vibration patterns.

SRS 03 – The gateway shall aggregate data from the vending machine and forward processed metrics to the cloud platform via MQTT/HTTPS every 5 minutes.

SRS 04 – The software shall maintain a local data buffer capable of storing 24 hours of sensor readings in case of communication failure, with automatic retransmission upon connection restoration.

Concept Refinement

The vending machine industry represents a $22-25 billion global market with a critical need for smart health monitoring systems, as traditional maintenance approaches are bleeding operators dry with hidden costs that eliminate profits for 73% of independent vending operators. Current maintenance expenses range from $100-500 annually per machine, with monthly checks costing $50-100 and service incidents adding $50-200 each time, creating a maintenance burden that constitutes up to 15% of total operational costs. VendGuard is a plug-and-play IoT solution that transforms maintenance from reactive to predictive, using sensors to monitor critical vending machine components in real-time—including power draw, temperature, dispensing motor function, and cooling system performance. By identifying potential failures before they occur, VendGuard reduces maintenance costs by up to 25%, prevents emergency repairs, optimizes service routes, and extends machine lifespan. For a mid-sized vending operator with 100 machines, VendGuard can provide annual maintenance savings of $15,000 ($600/machine × 100 machines × 25% reduction), while simultaneously increasing machine uptime, reducing product spoilage, and improving customer satisfaction.

Power Budget, Hardware and Software BOM

Link: https://docs.google.com/spreadsheets/d/1TTWYfVYijEID34pga0s4eGWvey3Sqpn2U45rw3sVCuY/edit?usp=sharing

Deliverables

Fleet management email alert Video:
https://drive.google.com/file/d/1wrj_DSsPnJNEvwQ6pkhyjpANsUDaOeGC/view?usp=sharing

OTA with LoRaWAN Video: https://drive.google.com/file/d/1f_kNWFBVPYnf2AP8PctA-6Hh7hO_H4uW/view?usp=sharing