StillCold

Environmental monitoring without opening the door

Problem context

Why this project exists

The challenge

- **Refrigerated or enclosed spaces** are often monitored by opening them to check temperature.
- Opening the environment **disrupts** the internal conditions and makes it harder to see what's really happening over time.
- Many solutions depend on **Wi-Fi or cellular** networks, which can be unreliable during power issues or add cost and complexity.

What this project solves

StillCold is motivated by a simple need:

> Measure temperature inside a refrigerated or enclosed space **without going inside** — and **without** relying on Wi-Fi or cellular.

- Use **Bluetooth Low Energy (BLE)** so a nearby phone or device can read the data.
- No need to open the environment; no need for internet or a cellular plan.
- Practical basis for learning how sensors and short-range wireless can support reliable, low-overhead monitoring.

System overview

Hardware and architecture

Hardware at a glance

Role	Components
Controllers	ESP32-C6, Arduino Nano (small computers that run the system)
Sensing	HTU21D temperature and humidity sensor
Safety	Logic level shifter (protects parts that use different voltages)
Rest	Breadboard, wires, USB power, basic enclosure

How the system is organized

- **Sensing component**
  Talks to the temperature/humidity sensor and collects readings on a schedule.
- **Communication component**
  Makes the latest readings available over **Bluetooth Low Energy (BLE)** so a nearby device can read them.
- **Data flow**
  Sensor → Sensing component → Communication component → BLE → phone or other device.

This split keeps responsibilities clear and makes the design easier to understand and change over time.

Sprint structure

Building StillCold in stages

Two sprints, clear goals

- **Sprint 1**
  Get a working end-to-end prototype: measure temperature and expose it via BLE so it can be read from a nearby device.
- **Sprint 2**
  Improve reliability and finish integration with a mobile app so users can see live temperature on their phone.

Sprint 1 weekly focus

Week	Focus
1	Hardware setup and checking that the sensor and controllers work together
2	Reliable temperature collection and moving data between components
3	BLE service and readable temperature data
4	End-to-end check and initial mobile app structure

Sprint 1 MVP — Definition of Done

What “done” means for the first prototype

Sprint 1 goal

Establish a **reliable end-to-end pipeline**: measure temperature inside the monitored environment and expose it via BLE so a nearby device can read it.

Definition of Done — Features

- **Internal temperature measurement**
  The system measures temperature inside the monitored space; humidity is collected as extra context.
- **Wireless access via BLE**
  The latest temperature is available through a BLE characteristic that a nearby device can read.
- **End-to-end data flow**
  Temperature flows from sensor → system → BLE without manual steps.
- **No Wi-Fi or cellular**
  The system works without internet or cellular connectivity.

Definition of Done — Requirements (summary)

- System periodically collects temperature from the sensor.
- System sends the most recent temperature to a BLE-capable device when requested.
- Temperature can be read **without opening** the monitored environment.
- Data is simple and readable; the focus is on correctness and reliability over speed or polish.

Learning with AI

How AI supports learning in this project

The two chosen Learning with AI topics

1. **Sensors and Data Acquisition**

   How physical measurements are converted into digital data

2. **Applications as System Interfaces**

   How software applications interact with external hardware

AI as a “second brain”

In this course, AI is used **with** the learner, not **for** the learner:

- A place to **offload context**, ask better questions, and work through uncertainty.
- **Not** a replacement for thinking — a support for understanding sensors, documentation, and system behavior.

Where it is used in StillCold

1. **Sensors and Data Acquisition**
   How physical measurements (temperature, humidity) become digital values: resolution, timing, and limitations. AI helps interpret datasheets and relate theory to observed behavior.

2. **Applications as System Interfaces**
   How the mobile app talks to the hardware over BLE: patterns, constraints, and design choices. AI helps reason about the app as part of the whole system, not just a standalone screen.

How it is documented

- Learning with AI is **explicit** in the process: documentation records how AI-assisted exploration influenced design and understanding.
- Evaluation focuses on **evidence of understanding** (explaining behavior, justifying decisions) rather than volume of AI usage.
- Goal: **build understanding**, not outsource it.

Risks and constraints

What could affect the project

Risks and constraints

- **Sensor behavior**
  Sensors are not perfect; readings can vary with environment and setup. The design accounts for this, with testing under realistic conditions.
- **Scope and timeline**
  Sprint 1 is a 4-week MVP. Priority is given to a working pipeline over extra features; some polish and refinement move to Sprint 2.
- **Learning curve**
  Embedded systems and BLE are new territory. Structured sprints, clear milestones, and AI-assisted learning help manage complexity.
- **Hardware dependency**
  Progress depends on having working hardware. Early weeks focus on verification so issues are caught soon.

Sprint 2 outlook

What comes after the MVP

Sprint 2 focus

- **Reliability and stability**
  More consistent sensor readings and BLE data exposure.
- **Mobile app integration**
  The Flutter app connects to StillCold via BLE and shows live temperature to the user.
- **Richer environmental context**
  Humidity may be used more explicitly alongside temperature.
- **Refinements from Sprint 1**
  Update strategies, data format, or internal structure based on lessons from the first sprint.

Sprint 2 success looks like

- BLE communication with the mobile app is working and stable.
- Users can see current temperature on their device without opening the monitored environment.
- The system stays within its original scope: clear responsibilities, no major redesign required.

Thank you

StillCold — *Environmental monitoring without opening the door*

Questions?