Wireless RF transmission system for a batteryless temperature logger

By

Stoyan Dinev &
Brian Joemmankhan

In collaboration with

BSc Electrical Engineering
EE3L11
Bachelor Graduation Project Electrical Engineering

TU Delft

Preface

This thesis was written as part of the Bachelor Graduation Project at TU Delft, within the Electrical Engineering program. It documents the design and development of a batteryless and wireless temperature sensing data logger, a solution aimed at applications such as cold-chain logistics.

We would like to express our sincere gratitude to our BAP supervisors, prof.dr. K.A.A. Makinwa and dr. Sijun Du, for their invaluable guidance and support throughout the project. We are also deeply thankful to our daily supervisors, Floris van Mourik and Teije Onstein, for their continuous support, technical insights, and encouragement during the development process. Special thanks go to Martin Schumacher for his essential assistance with component-related tasks. Finally, we are grateful for the collaboration and contributions of our fellow students and colleagues, including Kjell de Wit, Alec Reunis, Rolando Russel and Waris Ibrahimi.

Stoyan Dinev & Brian Joemmankhan
Delft, June 2025

Abstract

Global cold-chain logistics demands compact, maintenance-free data loggers that document storage conditions without adding batteries or e-waste. This thesis presents the wireless transmission subsystem of a fully batteryless platform that harvests energy, stores data locally, and transmits it through an NFC link. After researching wireless standards, an ISO 15693/NFC-Forum Type-5 solution built around STMicroelectronics' ST25DV64KC dynamic tag was selected for its I²C and RF dual-interface, non-volatile EEPROM and energy-harvesting capabilities.

The tag communicates with an ultra-low-power STM32U083 microcontroller over its I²C-bus to store the temperature data. Then, the data is transmitted to the reader via the connected antenna through the 13.56 MHz NFC electromagnetic field. At the same time, the antenna harvests residual energy from the reader, which is stored in an auxiliary capacitor. This stored energy provides a start-up power source for the energy harvesting system. Finally, a Python desktop script translates raw memory blocks into timestamped curves to visualise the logged temperature data.

Tests show that the NFC field of recent smartphones can provide sufficient energy for at least a full system duty-cycle in a single read. Additionally,the ST25DV64KC enables secure, wireless data retrieval , completing the architecture for integration into the project's complete batteryless logger. Improvements can be made by optimising the auxiliary capacitor storage, antenna design, possible readout by NDEF and the UI by thorough software design.

Contents

1 Introduction

1.1. Problem Definition

Globalisation has expanded access to diverse foods and pharmaceuticals year-round, but it has also extended and complicated supply chains. Longer transport routes and multiple handling stages increase the risk of temperature deviations, making cold-chain management essential to preserve the quality of perishables and temperature sensitive drugs from origin to destination [1].

Maintaining strict temperature control is critical in both food and pharmaceutical supply chains. In the case of perishables, even modest deviations can cause spoilage, premature ripening, chilling injury, or loss of firmness, leading to reduced shelf life and significant waste. For instance, bananas must be transported around 13.2–14.0°C to avoid chilling damage or accelerated ripening [2], [3]. Similarly, pharmaceuticals require storage within narrow temperature ranges, typically +2°C to +8°C. A vaccine monitoring study in Mexico revealed frequent overheating and freezing incidents during storage and transport, underscoring the widespread difficulty of maintaining proper cold-chain conditions [4]. Failures to maintain temperatures within specifications, in either sector, can lead to product rejection, economic losses, and in the case of pharmaceuticals to severe public health risks.

Continuous temperature data loggers are key to detecting and documenting excursions, yet most existing devices rely on batteries, requiring periodic replacement or disposal. Meanwhile, global e-waste is surging: 53.6 metric tons generated in 2019 (up 21% in five years) [5], and 62 in 2022 with projections of up to 82 metric tons by 2030 [6]. Therefore, there is a pressing need for loggers that combine dependable, continuous monitoring of perishables and pharmaceuticals with sustainable design: a batteryless, energy-autonomous, ultra-low-power wireless device with an operational lifetime beyond conventional battery-powered units would cut maintenance, lower total cost of ownership, and significantly curb e-waste.

1.2. Scoping and Bounding

The Problem Definition identifies key shortcomings in current loggers, such as battery dependence, frequent maintenance, and e-waste. These issues are used to translate stakeholder needs into clear objectives, constraints, and performance metrics. Logistics operators require unattended, reliable temperature monitoring; quality-control and regulatory bodies need continuous, tamper-evident records; sustainability officers demand minimal environmental impact and end-users must trust product integrity.

The objectives are: continuous and accurate temperature monitoring for perishables and pharmaceuticals; batteryless operation and autonomy over full shipment durations; wireless data retrieval; and a cost-effective design. Success criteria include approximately ±0.25°C accuracy(chosen to detect small deviations known to impact produce quality and pharmaceutical efficacy [2], [3], [4]), configurable sampling rate(e.g., 1 min to 1 h), uninterrupted operation for weeks to months(matching usual cold-chain shipment durations [3]) and reliable data delivery in cold-chain settings. Constraints cover limited ambient energy in refrigerated environments, minimal maintenance, and intermittent connectivity. The

design parameters to explore include sensor selection, microcontroller choice, energy-harvesting and -storage methods, and the wireless communication protocol. Evaluation will compare logged temperature against reference, harvested energy profiles versus consumption, transmission quality, uptime, and data completeness. These definitions guide architecture and component choices in later chapters.

1.3. State of the Art

Current market offerings such as Sensitech, ELPRO, and LogTag provide reliable temperature logging and often include wireless data retrieval or alerting [7], [8], [9]. These devices deliver proven accuracy and user-friendly interfaces, making them staples in perishables and pharmaceutical logistics. However, all depend on batteries, which limits their lifespan, necessitates periodic replacement, and contributes to maintenance overhead and electronic waste.

Academic research has explored batteryless, wireless sensing but typically falls short of comprehensive cold-chain logging requirements. Surveys of battery-less RFID-based wireless sensors highlight energy-harvesting approaches for intermittent sensing and communication, yet these tags generally offer only short interactions with a reader and have minimal onboard memory for extended storage [10]. Long-range batteryless RF sensors for produce freshness demonstrate that temperature measurements can be powered via RF harvesting, but they require close proximity to a reader and cannot store large sequences of readings over weeks or months [11]. Similarly, batteryless smart tags for fish freshness monitor temperature via RF energy harvesting but offer limited buffering and rely on reader presence [12]. LoRaWAN-based, battery-free sensor nodes designed for structural health monitoring illustrate that energy-autonomous wireless communication is feasible, yet these prototypes do not address the need for substantial local data storage over full shipment cycles [13]. Across these studies, energy harvesting and ultra-low-power communication techniques are promising, but memory capacity and autonomy duration remain insufficient for continuous temperature logging in cold-chain contexts.

No existing commercial logger or research prototype fully meets the combined requirements of battery-less operation, ultra-low-power, wireless communication, and ample onboard memory for long-duration, autonomous cold-chain temperature logging. This gap motivates the engineering of a new solution: a wireless, batteryless, ultra-low-power data logger capable of harvesting ambient energy in refrigerated environments, storing high-resolution temperature data over full shipment durations, and enabling reliable data retrieval without physical contact. This thesis contributes to that larger effort.

1.4. Project Objective

This thesis is part of a larger engineering project to design and prototype a fully batteryless, wireless, and ultra-low-power temperature logger tailored to cold-chain logistics. The system autonomously senses and stores temperature data throughout transport and enables contactless data retrieval at the destination. The device is composed of multiple subsystems:

1. **Energy harvesting:** Supply power from an external source.
2. **Temperature readout and storage:** Capture and store temperature readings.
3. **Wireless data transmission and display:** Wirelessly retrieve and display temperature data.

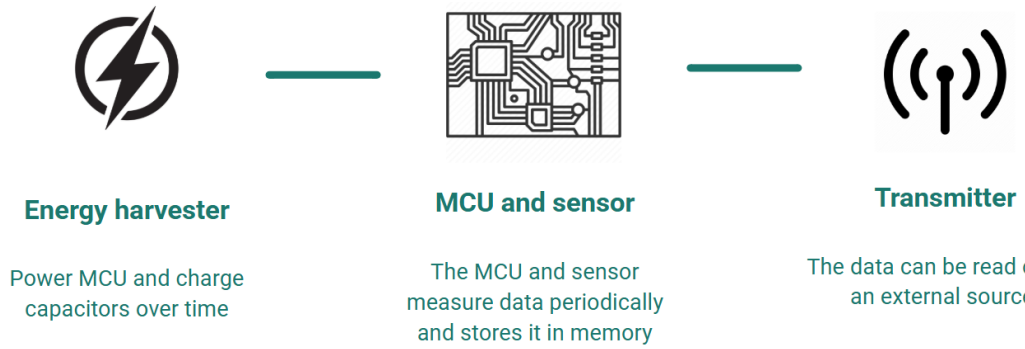

Figure 1.1: Diagram of full temperature logger system

This thesis focuses specifically on the third subsystem, the wireless transmission. It covers the selection of an appropriate transmission protocol, the corresponding hardware integration, and the development of a reliable data transmission interface. In addition, it includes a user interface for visualizing the logged data, presented as both a graph and a list of timestamped entries, accessible from a computer or smartphone.

1.5. Thesis Outline

The remainder of the thesis is structured as follows:

- **Chapter 2** presents the Programme of Requirements, distinguishing between system-level and subsystem-level constraints. These form the basis for the design and verification of the wireless temperature logging interface.
- **Chapter 3** details the design of the wireless subsystem. It motivates the selection of NFC Type 5 as the wireless technology, describes the ST25DV64KC NFC chip in detail, and outlines antenna selection.
- **Chapter 4** explains the system integration process. It first details the hardware connections between the microcontroller, energy harvester, and NFC memory. It then describes how temperature data is acquired, encoded, and stored, followed by an explanation of how this data is later retrieved and visualised.
- **Chapter 5** presents the results of practical system tests. It includes measurements of energy harvesting performance, verification of wireless data transfer reliability, and validation of the standalone Python interface used for data decoding and display.
- **Chapter 6** reflects on design choices, discusses limitations, and explores opportunities for improvement in energy storage, transmission range, and user interface design.
- **Chapter 7** concludes the thesis by reviewing the overall outcomes. It reflects on how closely the Programme of Requirements was followed and evaluates how effectively the project objectives were met.

2

Programme of requirements

The system outlined in the following chapter is a subsystem of the larger project, as elaborated in the previous chapter. To ensure the stand-alone readability of this document, the requirements for the complete system are separated from the subsystem as 'System Integration' and 'RF transmitter & memory' respectively.

2.1. System Integration Requirements

Mandatory (MR)

MR1.1 Must operate on 2.1V DC at an average power ≤2.02μW.

MR1.2 The device shall make no use of batteries.

MR1.3 Must be self-starting without pre-charged capacitors.

MR1.4 Powered entirely by self-harvested electromagnetic energy.

MR1.5 Functional in a standard 20ft shipping container (L:5.89m W:2.35m H:2.36m).

MR1.6 No parts required to be installed outside the container.

MR1.7 Fully operational in temperature range –30°C to+50°C.

MR1.8 Must operate without human interference for a minimum 4 weeks.

Trade-off (ToR)

ToR1.1 Device should remain compact and weigh ≤1kg.

ToR1.2 Remain operational for ≥5h without a functioning beacon.

ToR1.3 Adopt a modular design for easy troubleshooting and future expansion.

2.2. RF Transmitter & Memory Requirements

Mandatory (MR)

MR2.1 Store 4 weeks of hourly 16-bit temperature data in ≥ 10752bit memory.

MR2.2 100% retrieval of 4 weeks of logged data, transferring ≥ 672 samples per session at once.

MR2.3 Wireless protocol supporting the memory size from MR2.1.

MR2.4 Transfer data wirelessly to the reader.

MR2.5 Memory must retain data without power supplied to it for ≥ 4 weeks.

Trade-off (ToR)

ToR2.1 Display data as graph and table with timestamps.

ToR2.2 Read latency for a 16kbit log ≤ 2s.

ToR2.3 RF transmission minimum read range ≥ 4cm.

ToR2.4 RF transmission system shall be able to charge a 49 µF capacitor to 2.1V in ≤2s.

ToR2.5 Newest & performing RF transmission protocol, ex.: NFC: ISO/IEC 15693 + NFC Forum Type-5.

3

Wireless system design

3.1. Wireless technologies

Several wireless technologies were evaluated for the batteryless, ultra-low-power temperature logger: Bluetooth Low Energy (BLE), LoRa, passive RFID, and NFC Type 5 (ISO/IEC 15693 + NFC Forum Type 5). Each option was compared against the Program of Requirements: truly batteryless operation (MR1.2, MR1.4), strict power budget (MR1.1), autonomous operation for full shipment durations without human intervention (MR1.8), operation entirely inside a shipping container (MR1.5, MR1.6), sufficient throughput and memory retrieval for weeks of hourly logs (MR2.1, MR2.2, MR2.5), and straightforward contactless data retrieval (MR2.4).

BLE was considered due to its broad smartphone support and moderate-range capabilities. However, active BLE transmissions draw power in the tens of milliwatts range, which exceeds what can be harvested in refrigerated environments [14], [15]. General IoT protocol surveys also note that BLE's "low-power" still implies milliwatt-level active currents unsuitable for fully batteryless designs [16], [17]. Battery-free BLE prototypes depend on external power-transfer schemes not practical inside sealed cold-chain containers [14]. Moreover, BLE requires connection setup and maintenance (GATT/GAP), increasing firmware complexity and risking incomplete transfers when harvested energy is intermittent [18]. Therefore, BLE cannot meet the zero-battery, minimal-maintenance, and guaranteed uptime requirements.

LoRa offers long-range, low-data-rate communication and is well suited for continuous field deployments such as agricultural monitoring [19], [20]. IoT reviews highlight LoRaWAN's suitability for remote telemetry but also its significant transmit power spikes during packet transmission [17]. In principle, LoRa could support frequent or continuous transmission if energy were available, but its transmit power peaks far exceed the capability of ambient harvesting in refrigerated containers, making batteryless operation infeasible [21]. Additionally, LoRa relies on gateway infrastructure [22], conflicting with stand-alone, inside-container deployment (MR1.6). While LoRa would be attractive if continuous transmission were a PoR (e.g., live telemetry in open environments), it fails the energy-autonomy and infrastructure-free constraints for this cold-chain logger.

Passive RFID (UHF/HF) harvests energy from a reader field and can operate without onboard batteries. However, typical RFID tags lack sufficient onboard memory or integrated MCU capability to log hourly readings over weeks [12], [11]. Surveys of battery-less RFID-based wireless sensors confirm that passive tags can power minimal sensing during reader interrogation but generally offer only brief interactions and limited storage [10], [16], so they cannot autonomously accumulate large datasets over shipment durations or support bulk retrieval in one session (MR2.2). Although RFID avoids a local power source, its limited storage and dependence on reader presence during logging conflict with continuous, unattended operation.

NFC Type 5 (ISO/IEC 15693 + NFC Forum Type 5) is a specialized subset of RFID that aligns closely with the PoR. NFC tags can harvest energy from the reader field, supporting brief microcontroller activa-

tion for sensing or data transfer [10], [16]. Its very short read range (<10 cm) aligns with the workflow: the device remains dormant during shipment and is read out on-site by a worker at the destination, avoiding the need for continuous connectivity (MR1.6, MR1.8). NFC protocols support block read/write operations and larger memory configurations than simpler RFID tags, enabling transfer of weeks' worth of hourly temperature samples in one session (MR2.1, MR2.2). Integration via standard interfaces (e.g., I²C) allows seamless MCU connection. Compared to generic passive RFID tags, NFC Type 5 offers larger memory options, higher-level protocol support for block reads/writes, and broad smartphone compatibility for convenient data retrieval without specialized readers.

In summary, BLE and LoRa cannot satisfy the ultra-low-power, batteryless, and infrastructure-free requirements; passive RFID lacks the necessary memory and autonomy for multi-week logging; NFC Type 5 provides reader-powered energy harvesting, sufficient power for MCU bursts, adequate throughput for full-log transfer in a single session, and simple on-site retrieval. Accordingly, NFC Type 5 is selected for the wireless subsystem. The following sections detail the chosen NFC hardware, its integration with the microcontroller, and protocol implementation to fulfil the requirements.

3.2. NFC module choice

To determine the optimal NFC device for a batteryless and wireless temperature sensing data logger, a comparative analysis of several possible NFC devices was concluded. This section presents the research, criteria, evaluation and outcomes that led to the final selection of the ST25DVxxKC chips, which will now be abbreviated to ST25 from here and onwards in this thesis.

The methodology behind the research involved analysing technical data sheets, application notes and relevant scientific available technical literature for NFC tags. Brands such as STMicroelectionics, NXP, Onsemi and other brands were compared systematically based on the following criteria:

Table 3.1: Criteria for selecting the NFC chip

- Memory type (capacity and architecture)
- Power consumption
- Data transfer speed and throughput
- Wired an wireless connection type
- Energy harvesting capabilities
- Cost and supply chain factors

Memory capacity and architecture The first criterion is the memory type. Since the full system wakes up hourly, does it's measurements, saves the measurements and then goes back to sleep, choosing non-volatile memory is self-evident. Many NFC Tags include such memory so the tag can work separately from the rest of the system. Moreover, the MCU does not include sufficient non-volatile memory itself, thus a tag including memory is required so that the memory required no power in idle as per MR2.5.

Enough memory capacity is a crucial. Hourly temperature data of at least 4 weeks has to be stored in the memory the set sample size of the temperature sensor is 16-bits, per MR2.1, this will result in a minimal storage of:

$$4 \text{ weeks} \times 7 \text{ days/week} \times 24 \text{ hours/day} \times 16 \text{ bits/hour} = 10752 \text{ bits} \tag{3.1}$$

The ST25 has models of 4-, 16- and 64 Kbits of EEPROM (Electrically Erasable Programmable Read-Only Memory) programmable in 16-byte rows[23]. The N24RF64 also offers 64Kbit but required 4-page writes[24]. Additionally the M24SR64-Y matches the 64Kbit capacity[25]. NTAG 216 carries 888Bytes[26] and NTAG424DNA 416 Bytes[27] of user memory, both sufficient for multiple days of 16-bit sampling, but not the required 10752 bits from equation (3.1), ruling the NTAGs out. Additionally, the 10752bits requirement is fulfilled with the possible 16kbit variants, but 64kbit variants are chosen for expandability and no other disadvantages entailed with increased memory size.

Power consumption For batteryless operation, an idle power draw and minimum voltage as low as possible is to be desired. The tag has to be operable at ≤2.1V, per MR1.1, otherwise an additional voltage step-up circuit is required, which will draw additional power.

The ST25 only draws a maximum of 1.5 µA in a special 'low power down' stand-by mode , which will later be exhibited in chapter 3.3, a maximum of 160 µA during a read operation over I^2C [23, table 249] and 210 µA during a write, all at 1.8V. Moreover, ST's older M24SR64-Y can demand up to 5 µA in standby, 250 µA during read and 550 µA during a write, all at 3.3V[25, table 77]. Finally, onsemi's N24RF64 can require up to 10 µA during idle, 150 µA during a read operation and 400 µA during a write at 1.8V[24, table 3].

Since the write time for these three chips all equal $t_W = 5ms$ in a typical use case, the systems only logs temperature data hourly, writes it to the memory, and then goes into standby, the ST25 promises to be best of the three possible tags when it comes to power consumption. Namely, the ST25 consumes less power than the M24SR64-Y, while running at lower voltage, and also consumes less than the N24RF64 during a write cycle, which is the most important power metric. This will be revisited and elucidated in chapter 4.1.

Wired and wireless connection type Since the MCU communicates to external devices over I^2C , it is crucial to have I^2C bus capabilities. Conveniently, all three NFC tags use the I^2C protocol, which is used to write the hourly measurement data to the EEPROM. The ST25, uses a I^2C bus frequency of 1Mhz- 'Fast-mode-plus' and commits in 16-byte rows in 5ms. It's NFC protocol is based on ISO/IEC 15693, has NFC type 5 certification and has read access up to 53 Kbit/s.The M24SR64-Y's I^2C bus is limited to a frequency of 400 kHz. Its RF protocol is compatible with ISO/IEC 14443 Type A and NFC Forum Type 4 Tag and boasts a higher 106 Kbit/s data rate. The N24RF64 has the same capabilities as the ST25, 1Mhz I^2C bus frequency, RF transmission based on ISO/IEC 15693, and NFC type 5 certification with 53 Kbit/s Data Rate. Thus, while all three tags are advantageous, none is particularly outstanding,

Energy harvesting capabilities Energy harvesting is an optional yet exceptionally practical addition, chapter 4.2 will elucidate on why this is and ToR2.4. The ST25 claims a maximum energy-harvesting capacity in the mW range[28]. N24RF64E offers similar energy harvesting capabilities[29].The NTAG and M24SR families do not have a power out pin[25][26][27]. Accordingly, only the ST25 and N24RF64 support energy harvesting.

Cost and supply chain factors As this project is performed under the TUDelft, it is desired that all components are to be ordered from Farnell or RS for contractual reasons, thus all prices will be gathered from these sellers.

Present-day listings show a clear supply-chain gradient. The ST25 is the easiest buy: three package options (SO8, TSSOP8, UFDFPN8) are stocked with a combined ≥9 000 units available from €0,887–€2,290 depending on package and amount[30]. Farnell flags the line as "Active" with <9-week factory lead-time for larger call-offs. M24SR64-Y stock has declined to zero since the start of this research and factory lead-time is claimed to be an year since the production has been brought to a halt[31]. By contrast, N24RF64 appears in Farnell's catalogue only as "Non-stocked / MOQ-on-request" and low stock, around 800 units at RS components, the distributor quotes a 18-week factory lead for the TSSOP-8 option, with indicative pricing from 0,736 to €1,050 per unit[32]. In short, all three parts are contract-compliant, but only the ST25DV64KC offers both immediate -stock availability and pricing that scales smoothly from a couple to small production amounts, whereas M24SR64-Y is serviceable yet supply-constrained with extended lead-time, and N24RF64 carries procurement risk due to low-stock status.

3.3. ST25DVKC Analysis and description

The ST25 is a dual-interface NFC tag integrating an ISO 15693/NFC Type 5 RF transmission with a Fast-mode Plus I²C (1 MHz) wired interface.

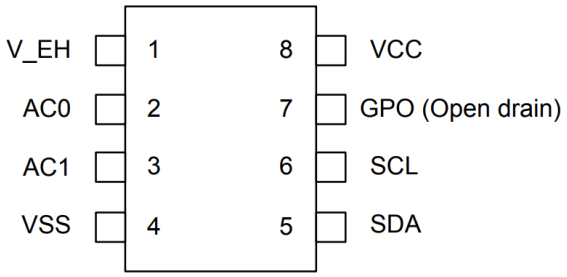

Figure 3.1: Pin layout of the ST25

The pin diagram and its functions, in figure 3.1, include: antenna terminals (AC0, AC1) for the RF interface, V_{EH} for harvested energy output, SDA and SCL for the I²C interface, the GPO for interrupt signalling and indicating incoming RF or I²C events, and V_{CC} and V_{SS} for power supply connections.

3.3.1. Power Management

The ST25 is able to operate with an external voltage supply ranging from 1.8 V to 5.5 V. Its energy harvesting capability, activated by enabling the EH_EN bit, allows the device to capture energy from the RF field of the reader, providing in the mW-range of power at the V_{EH} pin. Additionally, the tag is able to enter a low power down mode when prompted, with an extremely low typical current consumption of I_{CC} 0.84 µA at 1.8V, when neither RF nor wired power is present. The power is transferred to the ST25 by RF at 13.56 MHz via coupling antennas in the ST25 and the Reader.

3.3.2. Memory Organisation

The ST25DV64KC features a 64-kbit EEPROM memory, logically divided into four user-configurable memory areas and a separate system configuration area. The segmentation of the user memory is controlled using the end adresses, ENDA1 to ENDA3, of the areas, enabling flexible definition of memory regions in 32-byte increments. This allows for tailored allocation of space for NDEF message storage, sensor data logging, or reserved system use [23, table 4]. Each of the four user memory areas can be individually protected for read and/or write access using one of three dedicated 64-bit RF passwords. This facilitates secure, application-specific access control to the memory.

The user memory region of the ST25 begins with the Capability Container (CC) message. The CC is mandatory for NFC Forum Type 5 Tag operation and defines how the NDEF message area (T5T_Area) is mapped within the user memory. It resides at the very first bytes of the user memory and encodes key parameters such as the magic number, version/access conditions, MLEN (encoded memory length), and feature bits. In essence, the CC informs an NFC reader of the supported addressing mode, maximum NDEF area size, read/write access conditions, and optional features [33].

For the ST25DV64KC (64 kbit EEPROM), an 8-byte CC format is used. Depending on how much of the user memory is dedicated to NDEF storage and whether multiple-block read should be enabled, different CC values apply. For NFC Forum certification using the full user memory and with multiple-block read enabled, the CC file must be set to:

$$E2\ 40\ 00\ 01\ 00\ 00\ 03\ FFh \tag{3.2}$$

This value breaks down as follows(per AN4911 Section3 [33]):

- **Magic number = E2h**: indicates two-byte block addressing (required for ≥16 kbit memory).
- **Version and access conditions = 40h**: version 1.0, read/write always allowed for smartphone interoperability.
- **Bytes 2–3 = 00 01h**: encode part of T5T_Area sizing: here indicating full user memory is used.
- **Additional feature info and RFU bits**: set to enable multiple-block read without errors across area borders.

- **MLEN = 03FFh**: for full 8192-byte user memory minus 8 bytes CC → (8192 – 8) / 8 = 1023 decimal = 03FFh. For NFC Forum certification this precise MLEN is required.

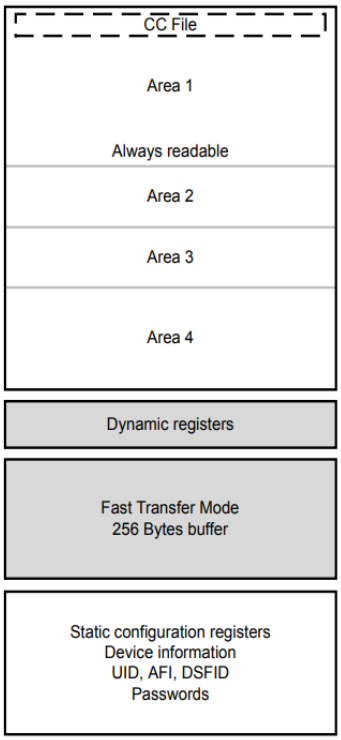

Figure 3.2: Memory Organisation

After the user memory overview and CC setup, the remaining memory-related components are:

- **Dynamic Registers**: Located at addresses 2000h–2007h in the dynamic memory map, these registers are accessible via both RF and I²C interfaces. They provide real-time status information and enable temporary activation or deactivation of certain device features [[23, table 14]. When a valid change is made to a static configuration register—whether via RF or I²C—the relevant dynamic register is automatically updated to reflect the new operational state.
- **Fast Transfer Mode Buffer**: The device provides a 256-byte fast transfer buffer that serves as a high-speed data mailbox between the RF and I²C interfaces. This feature is particularly useful for applications requiring efficient communication between contactless and wired domains.
- **System Configuration Area**: This dedicated area, located at addresses 0000h–0017h, contains static registers used for device configuration, password management, and operational mode settings [23, table 13]. Access to this area is secured by a 64-bit configuration password. It also stores essential, read-only information such as the device's IC reference, memory size, revision number, and a unique 64-bit identifier (UID), programmed by STMicroelectronics during manufacturing in compliance with ISO 15693 standards.

3.3.3. Data Protection

The ST25 offers security features with four independent 64-bit RF passwords controlling read, write, I²C protection, and system-level configurations. Additionally, it brings a separate 64-bit password securing the I²C interface. Various access levels such as read-only, write-once, or complete lockout can be configured and protected with these passwords. Unauthorized access will result in session closure until a reset or correct authentication. The open-drain GPO pin will also signal these key events, including RF field detection, password authentication failures, and mailbox activities, providing immediate notification to the host MCU.

3.3.4. Fast Transfer Mode (FTM)
Although the ST25 provides a 256-byte mailbox for high-speed data exchanges between RF and I²C interfaces, the data transmission requirement of 10752 bits per transmission, according to MR2.1 exceeds mailbox capacity. Therefore, conventional multi-block read and write operations will be utilised instead of FTM.

3.3.5. RF Operation Overview
The ST25 follow ISO/IEC 15693 or NFC Forum Type 5 standards which meets ToR2.5. The device communicates via he 13.56 MHz carrier electromagnetic wave, demodulating incoming data from amplitude shift keying (ASK) modulation. These incoming ASK signals either use 10%, or 100% modulation depth, with data rates of 1.6 kbit/s using 1/256 pulse coding mode, or 26 kbit/s using the 1/4 pulse coding mode. Outgoing data is generated via load modulation and Manchester coding, using sub-carrier frequencies of 423 kHz and 484 kHz. This achieves data transmission rates of 6.6 kbit/s in low data rate mode, and 26 kbit/s or 53 kbit/s in high data rate mode, with both high data rate modes using a single sub-carrier frequency of 423 kHz.

3.4. Antenna Design
The ST25 tag IC provides dual-interface functionality, allowing I²C communication with a host microcontroller. However, RF operation at 13.56 MHz requires an inductive coil(antenna) to interface with an NFC reader. The antenna serves two primary purposes: (1) enabling data exchange through load modulation when exposed to the reader's magnetic field, and (2) harvesting energy from that field to charge a capacitor in the ultra-low-power, batteryless temperature sensor. Without a properly tuned antenna, the tag cannot reliably resonate at 13.56 MHz, leading to degraded read range, unstable communication, or insufficient harvested power [34].

3.4.1. NFC Antenna Design Principles and Formulas
Fundamental design principles for an NFC tag antenna derive from its equivalent circuit fig.3.3 and are detailed in ST application notes AN2972 and AN5605 [34], [35]. The antenna coil is modeled as an inductance L in series with a parasitic resistance R_p. Resonance is achieved by combining this inductance with a total tuning capacitance C_{tun}, provided by the ST25 internal capacitance bank (nominal 28.5 pF) or, if required, by an external capacitor [23].

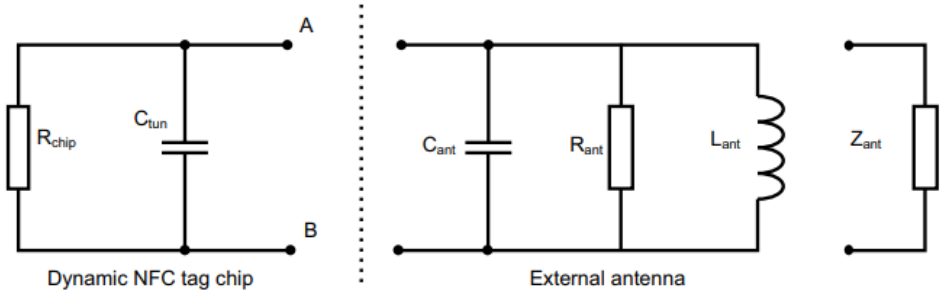

Figure 3.3: Equivalent circuit of the dynamic NFC tag chip and its antenna

The resonance frequency satisfies

$$f_{\text{res}} = \frac{1}{2\pi\sqrt{L\,C_{\text{tun}}}}, \tag{3.3}$$

as presented in AN2972[34]. At resonance, reactive impedances cancel, maximising coil current for both communication and energy harvesting.

The quality factor Q of the coil is defined by

$$Q = \frac{\omega L}{R_p}, \quad \omega = 2\pi f_{\text{res}}. \tag{3.4}$$

A moderate Q is required: excessive Q yields a narrow bandwidth and increased sensitivity to detuning from manufacturing tolerances or environmental influences (e.g., nearby metal, enclosure effects),

whereas insufficient Q reduces coupling efficiency and harvested power [35].

An antenna operating at 13.56 MHz can be designed in various shapes, depending on the specific application requirements. As previously discussed, the key parameter to consider is the antenna's equivalent inductance, L, at 13.56 MHz. Although the stray capacitance is challenging to estimate, it typically falls within a few pF for most NFC/RFID applications. For certain antenna geometries, AN2972 provides useful formulas to calculate the self-inductance, L, even without an estimate of the antenna's stray capacitance. For a square coil, one common approximation is

$$L \approx 2.34\,\mu_0\,N^2\,\frac{d_{\text{avg}}}{2+1.75\,\rho}, \quad d_{\text{avg}} = \frac{d_{\text{out}}+d_{\text{in}}}{2}, \quad \rho = \frac{d_{\text{out}}-d_{\text{in}}}{d_{\text{out}}+d_{\text{in}}}, \tag{3.5}$$

where d_{out} and d_{in} denote the outer and inner dimensions of the square coil (in metres), N is the number of turns, and $\mu_0 = 4\pi \times 10^{-7}\,\text{H/m}$ [34]. For higher inductance within a limited footprint, a double-layer coil may be implemented: top and bottom coils interconnected by vias effectively increase total inductance. ST application note AN5605 describes the design methodology for such double-layer antennas at 13.56 MHz [35], including recommended trace widths, spacing, turns, and interlayer connections, often using ST's eDesignSuite tool for simulation and optimization [36].

Once an estimate of L is obtained, the required tuning capacitance is computed by

$$C = \frac{1}{(2\pi f)^2\,L}, \tag{3.6}$$

and must be compared against the internal tuning range of the ST25 (nominal 28.5 pF). If C falls outside this range, provision for a small external capacitor is necessary. Coupling factor k between reader and tag coils depends on relative coil sizes, orientation, and separation. Since smaller coils exhibit lower coupling and shorter read range, coil geometry must satisfy the minimum read distance requirement (4 cm) specified in the Program of Requirements, ToR2.3.

3.4.2. Selection of the Commercial Antenna (ANT7-T-25DV64KC)

Antenna design primarily affects the achievable read range and the amount of energy that can be harvested by the NFC chip. Given that the Program of Requirements treats read range and energy harvesting as trade-off parameters(ToR2.3, ToR2.4), and the specified read distance does not impose stringent constraints for this application, no custom antenna was developed. Instead, the ST ANT7-T-25DV64KC 14 mm × 14 mm reference antenna board has been adopted because it is commercially available and specifically designed by STMicroelectronics to match the ST25.

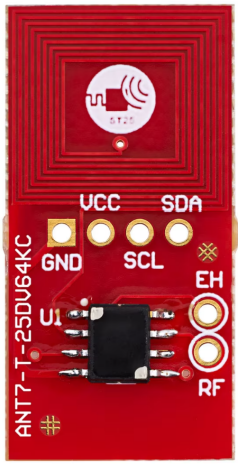

Figure 3.4: Top view of the ANT7-T-25DV64KC 14 mm x 14 mm double layer antenna reference board

Since ST designed both the antenna layout and the chip, compatibility and adequate performance can be assumed without further custom tuning efforts. The ANT7 breakout board offers a pre-tuned

double-layer coil yielding resonance at 13.56 MHz with the internal capacitance of the ST25, and provides accessible I²C test points and an energy-harvesting output for evaluation [37]. This approach avoids additional design iterations and measurement cycles while ensuring that read range and harvested power meet the system needs under the conditions defined in the PoR. If future requirements demand a different range or improved harvesting, a custom antenna design could be considered using the principles described in the section above, but for the present scope the ST ANT7-T-25DV64KC reference antenna is considered fully suitable.

4

System integration and User interface

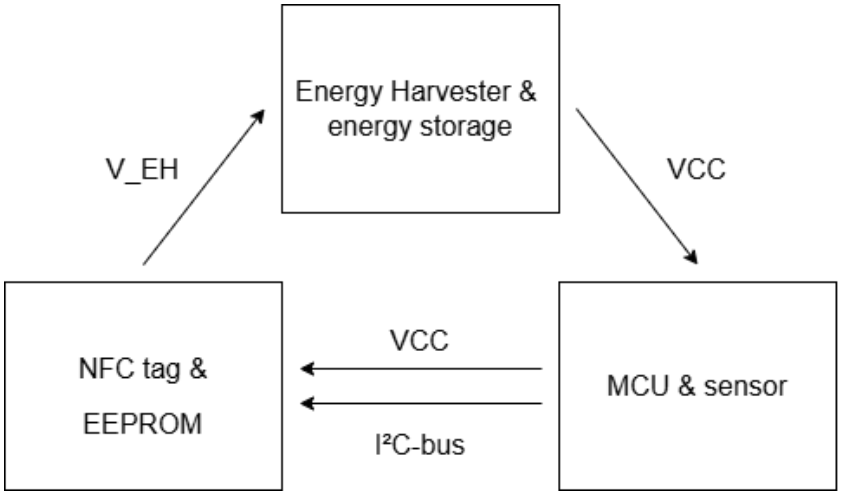

Figure 4.1: Simplified block diagram of the complete system

Figure 4.1 provides a simplified and functional block diagram including connections. A smartphone connects to the the ST25 NFC tag, which not only stores temperature data, but also harvests RF energy through NFC. This harvested energy is buffered and routed to the main energy harvesting circuit. The main energy harvesting circuit generates the regulated V_{CC} rail, powering the microcontroller and its sensor. During normal operation the MCU sends 16-bit temperature data over the shared I^2C bus to the ST25's EEPROM on hourly basis.

The sections that follow describe the interaction between the microcontroller, energy harvester, and ST25 NFC tag, explain how the system manages power during startup, and detail how temperature data is stored, transmitted, and processed for user access through a custom software interface.

4.1. MCU and ST25 connection

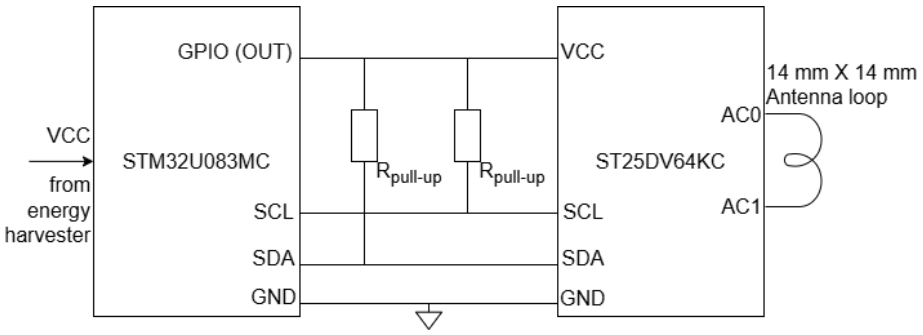

Figure 4.2: Schematic with all connections between the MCU and the ST25.

The MCU communicates with the ST25 over a two-wire I^2C bus, using Serial Clock (SCL) for synchronisation and Serial Data (SDA) for transfers of information. Both lines are held high by pull-up resistors so that they raise the I²C lines back to 'high', because open-drain devices can only pull them down. After each measurement cycle, the MCU puts the voltage to zero on the GPIO output to disable the connection between the ST25 and the MCU. This is possible due to the non-volatile EEPROM in the ST25 and will minimise stand-by loss.

4.2. MCU and energy harvester connection

The energy harvester powers the MCU via its regulated VCC rail. After a long idle period, such as a first-time startup, the MCU and its peripherals briefly draw a large inrush current. If the harvester's capacitors are not charged sufficiently yet, this surge can drop the rail below the 1.7 V brown-out threshold voltage and force the MCU to reset. This voltage dip may cause the MCU to reset and repeat the cycle, which is a serious risk that could cause an infinite boot-loop and prevent proper system operation.

4.3. ST25 and energy harvester system connections

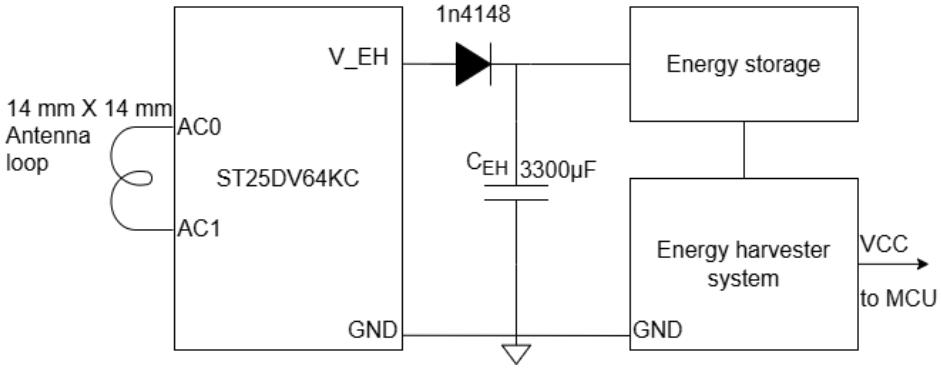

Figure 4.3: Schematic with all connections between the ST25 and energy harvesting system.

The ST25 is able to enhance the reliability of the energy harvesting system with supplementary RF energy harvesting whenever a NFC reader is held close to the device. The harvesting capabilities are integrated into the system through the ST25's dedicated V_ EH pin. The antenna captures energy from the 13.56 MHz electromagnetic field and the tag redirects the unused energy, charging an auxiliary capacitor C_{EH} to at least 2.29V, as later proven in chapter 5.1. An 1N4148 diode[38] is used between the ST25 and the auxiliary capacitor to ensure unidirectional energy flow, preventing reverse discharge. This specific diode was chosen due to its availability, and was beneficial because of its low voltage drop at low current flow.

Initially, a capacitor size of around 50 µF, was used storing at least approximately $E = \frac{1}{2}CV^2 = \frac{1}{2} \cdot 50 \times 10^{-6} \cdot 3.29^2 \approx 0.27$ mJ,when fully charged with a smartphone far within 2 seconds. This was determined to provide sufficient energy for a complete operational start-up cycle of the MCU. This means booting, measuring temperature, sending, storing the data in the memory, and idle for the rest of the hour. Thus ideally, the ST25 charges this capacitor in under 2 seconds to at least 1.8V, which is the 1.7V brown-out voltage plus 100mV safety margin, to meet ToR2.4.

However, a larger 3300 µF capacitor was chosen in to replace the smaller one due to its availability, significantly higher energy storage capability with longer charge retention, thus providing a more robust and reliable ”jump-start” for the system. If the larger capacitor were to be fully charged, the total energy stored is $E = \frac{1}{2}CV^2 = \frac{1}{2} \cdot 3300 \times 10^{-6} \cdot 3.29^2 \approx 18$ mJ. Moreover, to confirm that the larger capacitor acquires enough energy by charging just 2 seconds, as per ToR2.4, voltage measurements over the capacitor were performed, shown in chapter 5.1. Both the smaller 50 µF and the 3000 µF capacitor exist on the board of the prototype and can be swapped out using jumper-caps.

Additionally 1N4148 diode[38] is used between the ST25 and the auxiliary capacitor to ensure unidirectional energy flow, preventing reverse discharge.This specific diode was chosen due to its availability, and was beneficial because if its low voltage drop at low current flow.

The auxiliary capacitor C_{EH} temporarily stores the harvested energy, enabling voltage stabilisation above the aforementioned 1.7V during the MCU power-up phase. Once past that phase, the energy harvester can reliably sustain full operation independently.

4.4. Data transmission and readout

As explained above, the ST25 NFC chip serves both as non-volatile storage and as the wireless transmission interface. Each hour, the microcontroller awakens the system, acquires a 12-bit temperature measurement from the TMP75B sensor, and stores it in the ST25’s EEPROM as a 16-bit word. Since each memory block on the ST25 holds two 16-bit entries, every block contains two temperature readings.

To extract logged data, the ST25 NFC Tap application (iOS and Android) is employed [39]. The user launches the app, establishes a connection with the ST25, and navigates to the Tag Memory feature. Key operations from the software application include modifying the Capability Container (CC) file, erasing all memory, and reading memory contents. For data readout, the user specifies a start block (avoiding the first two blocks reserved for the CC file) and the number of subsequent blocks to retrieve. The application then saves the selected region as a binary file (.bin) on the smartphone, encapsulating all stored temperature logs.

The resulting .bin file is transferred to a host computer, where a Python-based executable processes it. Prior to describing the software, it is necessary to explain how the TMP75B sensor encodes its measurements and how the microcontroller reformats them for storage.

4.4.1. Temperature Sensor Data Encoding in Memory

The TMP75B outputs each measurement as a 12-bit two’s complement value left-justified in a 16-bit register: bits [15:4] hold the signed count in Q4 format (LSB = 0.0625°C) and bits [3:0] are zeros, so the raw sensor output is $bbbb\ bbbb\ bbbb\ 0000$ [40], [41]. Upon I²C read, the MCU combines the two bytes into a 16-bit word and performs an arithmetic right shift by 4 bits, yielding $0000\ bbbb\ bbbb\ bbbb$. This signed 12- bit value in a 16-bit container is stored directly in the EEPROM of the ST25, preserving the exact two’s complement Q4 reading for later conversion [40], [41].

Before examining concrete examples, note that to recover the original signed 12-bit count on the host, the stored right-justified word is first masked to 12 bits, then shifted left by 4, so that the sign bit moves into bit 15 for proper sign-extension. After sign-extension, a right shift by 4 returns the signed 12-bit count in the low bits, which is then multiplied by 0.0625 to yield the temperature in °C.

1. **Positive temperature example(+50°C)**
    - TMP75B computes 50 ÷ 0.0625 = 800 decimal → 320h hex, as a 12-bit pattern (0011 0010 0000). Left-justified, sensor output is 3200h (0011 0010 0000 0000).

- MCU arithmetic-right-shifts 3200h by 4 bits → 0320h (0000 0011 0010 0000) and stores this in EEPROM.
- Host reads raw word = 0320h, masks to 12 bits → 320h, shifts left 4 → 3200h, sign-extends (MSB=0 so remains 3200h), shifts right 4 → 0320h (800 decimal), then 800 × 0.0625 = 50.0°C.

2. **Negative temperature example(-25°C)**
   - TMP75B computes |–25| ÷ 0.0625 = 400 decimal → 190h as 12-bit (0001 1001 0000), then two's complement → E70h (1110 0111 0000). Left-justified, sensor output is E700h (1110 0111 0000 0000).
   - MCU arithmetic-right-shifts E700h by 4 bits → FE70h (1111 1110 0111 0000) and stores this in EEPROM.
   - Host reads raw word = FE70h, masks to 12 bits → E70h, shifts left 4 → E700h, sign-extends (MSB=1 so becomes signed16 = E700h – 10000h = –1900h), shifts right 4 → signed12 = –190h (–400 decimal), then –400 × 0.0625 = –25.0°C.

4.4.2. Python Script Processing and Visualization

To facilitate readout of the recorded temperature data, a custom Python script (Appendix B) was developed. While the code is not discussed line-by-line here, it closely follows the data format and reconstruction steps outlined in Section 4.4.1. Specifically, it interprets each 16-bit word as a packed 12-bit Q4 temperature sample, applies the appropriate masking, shifting, and sign-extension operations, and converts the result to a temperature in degrees Celsius. This tailored approach ensures full compatibility with the TMP75B sensor's encoding format and preserves the accuracy defined by its 0.0625 °C resolution [40].

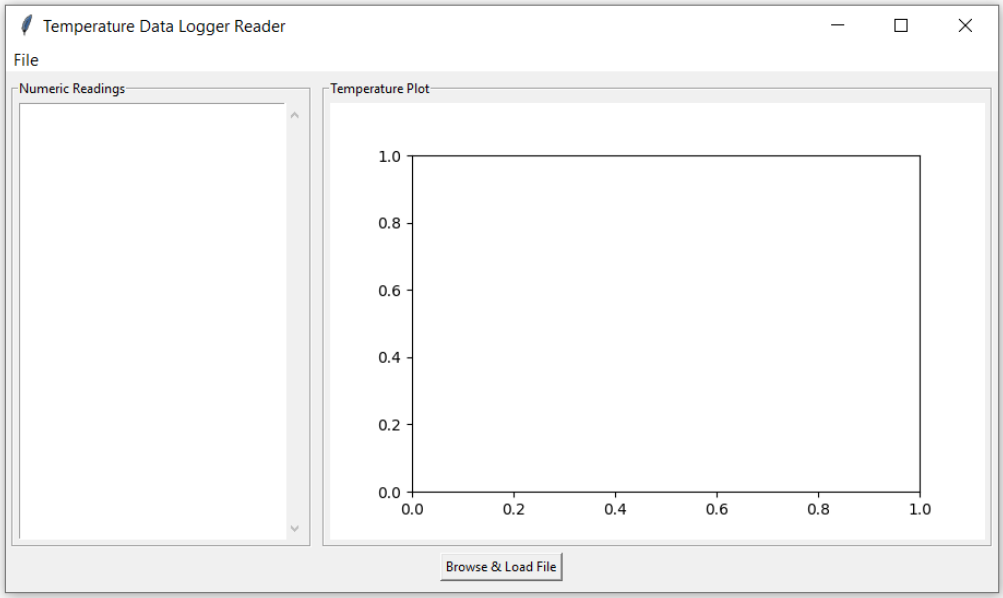

Figure 4.4: Graphical User Interface of the temperature data logger

The script provides a graphical user interface (GUI) with two primary features: a scrollable numeric table and a live-updating temperature plot. The table lists each measurement by hour, grouped under daily headers. The plot displays temperature over time, with the x-axis adapting based on dataset length, labelling either hours (if fewer than 24 samples) or days (in 24-hour blocks). Temperature values retain their original Q4 precision and are shown on the y-axis in °C. Together, the interface allows users to visualise long-term trends while retaining access to fine-grained data values.

To ensure portability and ease of use, the script was converted into a stand-alone executable using the PyInstaller tool. This process bundles all necessary Python dependencies into a single .exe file that can be run on any Windows machine without requiring a separate Python installation. As a result, end users can launch the application simply by double-clicking the file with no setup or command-line interaction being necessary. This approach enables broader accessibility for non-technical users while preserving the full functionality of the interface.

5

Results

5.1. Energy harvesting with the ST25 and smartphones

The ST25 proves to provide substantial energy over the V_ EH pin when a smartphone is held close to the antenna. Below is the gathered data that proves the reliability of the energy harvesting capabilities of the ST25. Tests were performed with both an iphone 15 Pro(iOS) and a Oneplus 13(Android) phone. Pictures of the measurement setup can be found in Appendix A.1, voltage measurements were performed with the EDU34450A digital multimeter[42].

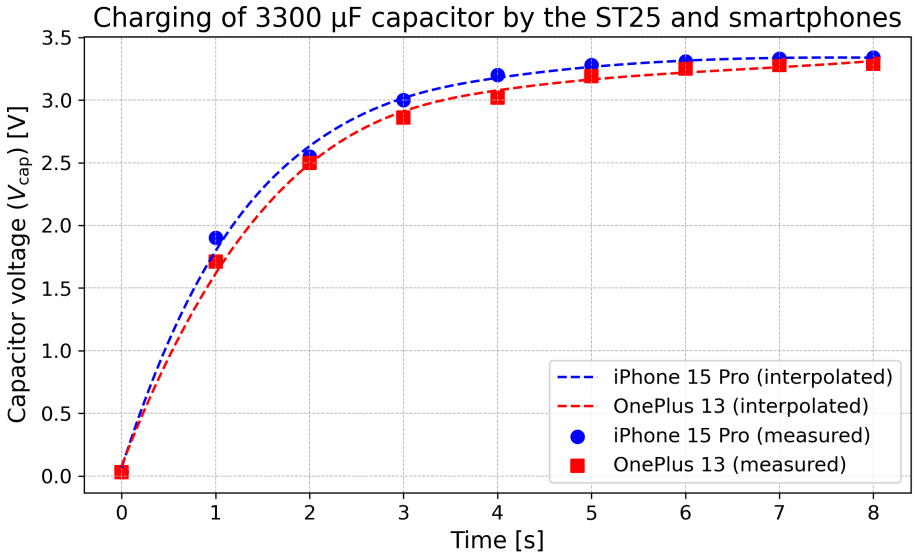

Figure 5.1: Capacitor voltage when charged by the ST25 and smartphone as reader compared to theory.

As seen in figure 5.1 both smartphones charge the 3300 µF capacitor quickly, with both reaching a voltage higher than 2.5 V in 2 seconds, a time that is reasonable to expect the user to hold the reader to the tag. At just 4 seconds the capacitors is around 90% chaged. The plot includes curves that are interpolated from the measured data points, the code can be found in Appendix B.2

5.2. Data transmission and visualisation

A key objective of this thesis was to verify that logged temperature data can be accurately stored, transmitted, and visualised using the developed wireless system. To validate this, two representative

.bin data files were created and uploaded to the EEPROM of the ST25 NFC chip via the ST NFC Tap App. Each binary file contained temperature readings encoded using the 12-bit Q4 format described in Section 4.4.1, and values were pre-shifted 4 bits to the right to replicate how the microcontroller stores them in memory. The first test file contained 8 samples (Table 5.1), and the second contained 700 samples, chosen to exceed the 672-sample minimum defined in the Program of Requirements (MR2.1, MR2.2).

Table 5.1: 12-Bit Q4 Example Data

Temperature	**Digital Output**	
	Binary	**Hex**
$127.9375\,^{\circ}\mathrm{C}$	0111 1111 1111 0000	7FF0
$125\,^{\circ}\mathrm{C}$	0111 1101 0000 0000	7D00
$25\,^{\circ}\mathrm{C}$	0001 1001 0000 0000	1900
$0.0625\,^{\circ}\mathrm{C}$	0000 0000 0001 0000	0010
$0\,^{\circ}\mathrm{C}$	0000 0000 0000 0000	0000
$-0.0625\,^{\circ}\mathrm{C}$	1111 1111 1111 0000	FFF0
$-25\,^{\circ}\mathrm{C}$	1110 0111 0000 0000	E700
$-40\,^{\circ}\mathrm{C}$	1101 1000 0000 0000	D800

To test data integrity and retention, each file was uploaded while avoiding the first two EEPROM blocks reserved for the Capability Container (CC) file. Multiple readout operations were then performed using a smartphone (both Android and iOS) to retrieve the data. In all tests, the NFC Tap App successfully extracted the .bin file, with no detected corruption or data loss. The 8-sample file was consistently read in under 1 second, while the 700-sample file required approximately 4 to 5 seconds per session. Although this exceeds the Trade-off requirement of under 2 seconds for 16 kbit data sizes(MR2.2), it still demonstrates acceptable latency for full memory retrieval and practical use.

The final step was to assess the performance and correctness of the stand-alone Python executable developed for data interpretation and display. Both binary files were processed using the executable, which launched reliably, though with a brief loading delay. As shown in Figure 5.2, the application accurately rendered the 8-sample dataset as an hourly temperature graph and correctly displayed the individual values in the adjacent table, with full consistency between the graphical and tabular outputs.

The larger 700-sample file, shown in Figure 5.2, confirmed that the software dynamically switches the time axis from hours to days once the number of samples exceeds 24. The temperature values retained their original precision, with correct decoding and formatting as specified by the TMP75B's Q4 encoding. While only the graphs are shown in the main text, the numerical output from the 8-sample and a fraction from the 700-sample files is provided in Appendix C. This confirms compliance with ToR2.1, which requires the data to be displayed as both a graph and a table with timestamps.

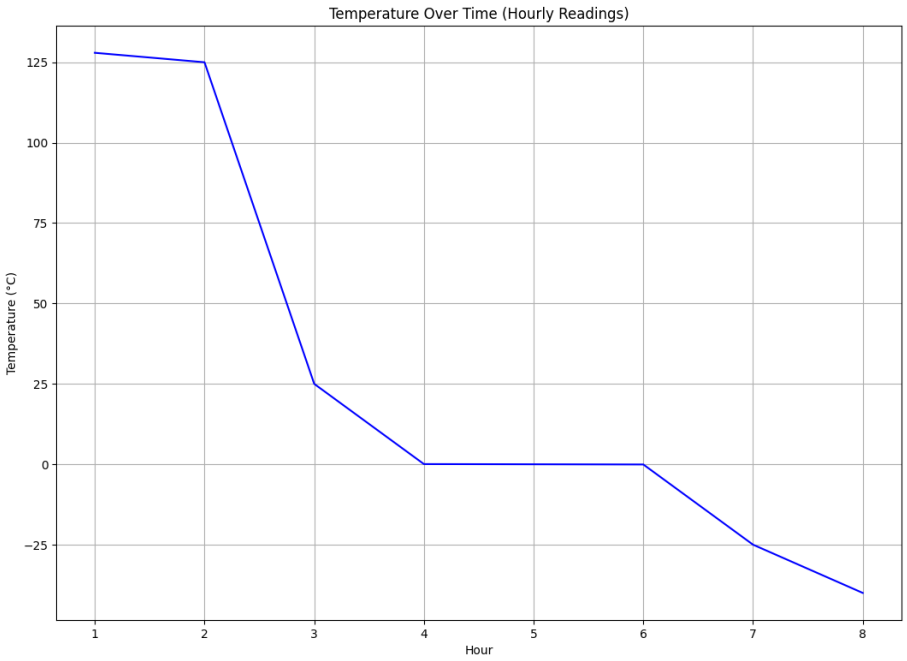

Figure 5.2: Graph from the user inteface showing the 8 hourly temperature readings from Table 5.1

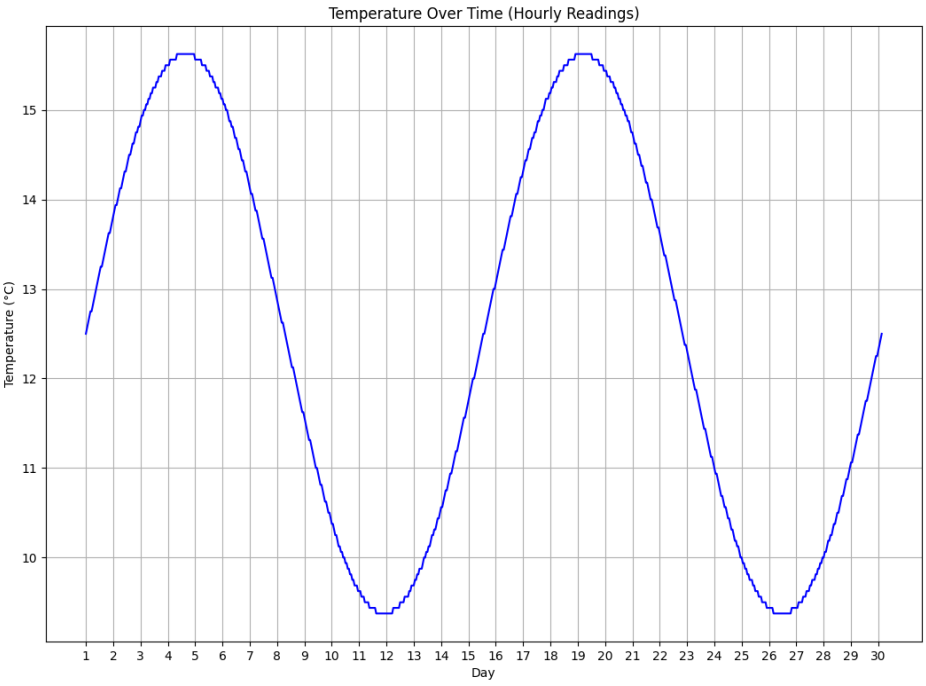

Figure 5.3: Graph from the user interface showing 700 hourly temperature readings

6

Discussion

Although all mandatory requirements from the Programme of Requirements for the wireless transmission system were satisfied, including most of the trade-off criteria, several aspects of the design deserve further discussion in terms of potential improvements or alternative approaches.

First to consider is the memory allocation of the NFC chip. According to the Programme of Requirements (MR2.1, MR2.2), the target memory size was at least 10,752 bits. The ST25DVxxKC family offers 4 kbit, 16 kbit, and 64 kbit variants. Although the 16 kbit version would have met the specification and reduced component cost, the 64 kbit option was used, for expandability and because it was already installed on the ANT7 breakout board. During development and testing, no drawbacks arose from using the larger memory capacity, so the 64 kbit device remained in place. However, in a production scenario where cost minimisation is a priority, the 16 kbit variant would be sufficient and more economical without affecting system functionality.

The antenna design also deserves consideration. Although initial research on antenna principles for the ST25 chip was conducted, no custom antenna was developed because the commercially available ANT7 breakout board and its built-in antenna were used. In testing, the read range between smartphones and the antenna came close to reaching 4-5 cm (ToR2.3). A more optimised antenna design or an alternative geometry could potentially extend the read/write range to around 10 cm, which would improve usability by reducing alignment sensitivity and enhance energy harvesting for charging the auxiliary capacitor.

In the scope of this project, the observed read/write distance was considered adequate, and the ST25's energy-harvesting pin delivered sufficient power under these conditions, so no bespoke antenna was pursued. If time had permitted, a tailored antenna design matched to the application would have been developed to maximise performance.

Thirdly, the current system design fulfils the energy requirements gracefully. Nevertheless, several refinements could enhance the performance. For example, substituting the 1N4148 diode with a low-drop Schottky diode could increase the auxiliary capacitor's peak charging voltage toward the tag's open-circuit output level, thereby boosting the percentage of usable harvested energy. Exploring intermediate capacitor values between 50 to 3000 µF could also be beneficial, since this could optimise the trade-off between charge time and stored energy.

Moreover, evaluating the system's Thevenin equivalent from the measured voltage-current curve would allow accurate digital simulation of the charging process, namely using the formula:

$$V_C(t) = V_{\text{th}}\left(1 - e^{-t/(R_{\text{th}}C)}\right). \tag{6.1}$$

Where time constant τ, denotes the time to reach 63% of the final voltage and directly informs the choice of a capacitor that balances rapid charging with sufficient energy storage in simulation.

Finally, the wireless transmission interface and the user interface warrant discussion. For data extraction from the ST25 device, the NFC Tap application provided by STMicroelectronics was used.

Although the app supported the required functions, it proved somewhat unreliable in practice, occasionally crashing or failing to respond to certain commands. Since its role was limited to verifying that a connection could be established between the NFC chip and a smartphone, and confirming data integrity throughout transmission, it was deemed adequate for testing. An alternative considered was the development of an NDEF (NFC Data Exchange Format) message. NDEF is a standardized format supported by NFC Type 5 devices and in principle could embed logic to perform actions without relying on external applications. For example, the device could convert stored temperature measurements from binary to decimal and trigger an email or SMS when tapped, bypassing the NFC Tap app.

However, implementing such NDEF-based functionality would have required substantial development effort and was outside the scope of this thesis, which focused on developing and verifying the wireless transmission system. Consequently, no NDEF message was implemented. In parallel, a Python-based graphical interface was created to read and display the extracted data. While the GUI provided the necessary visualisation, its Python implementation resulted in longer load times compared to a compiled application. A C++ executable would likely improve start-up speed, but the project team did not have sufficient expertise in C++. Given that full user interface development was not the primary objective, the existing GUI was acceptable for validation purposes.

The use of the NFC Tap app together with the Python script indicates that, for a market-oriented product, a dedicated mobile application for iOS and Android could be developed to handle NFC interactions, data processing, and user feedback in a seamless manner.

7

Conclusion

Through systematic research the ST25DV64KC NFC tag was selected to satisfy all essential system criteria and address nearly all performance trade-offs. Its I^2C and RF dual-interface, energy-harvesting capability, and non-volatile memory provided headroom for integration and reliable operation.

The tag's RF harvesting mechanism is able to charge an auxiliary capacitor. This stored energy supports the energy harvesting system, which then powers the microcontroller, enabling batteryless operation. Temperature readings are recorded and stored in the tag's non-volatile memory and retrieved with full accuracy. A supporting software tool decodes data after NFC transmission to the reader and updates the user display, validating the end-to-end system.

The energy harvester, sensing electronics, memory, and user interface create a complete system for monitoring perishable goods and pharmaceuticals. Although all mandatory requirements and most Trade-off requirements are met, a few targeted refinements, such as selecting a lower voltage drop diode, optimising auxiliary energy storage and antenna design, could optimise the system. These changes would enhance overall energy efficiency slightly, but not improve the overall operation of the system to a substantially significant amount. Thus, the final design provides a strong foundation for a state of the art, scalable, wireless and batteryless temperature data logger for perishable goods and pharmaceuticals.

A Pictures

A.1. Energy harvesting with the ST25DV64KC

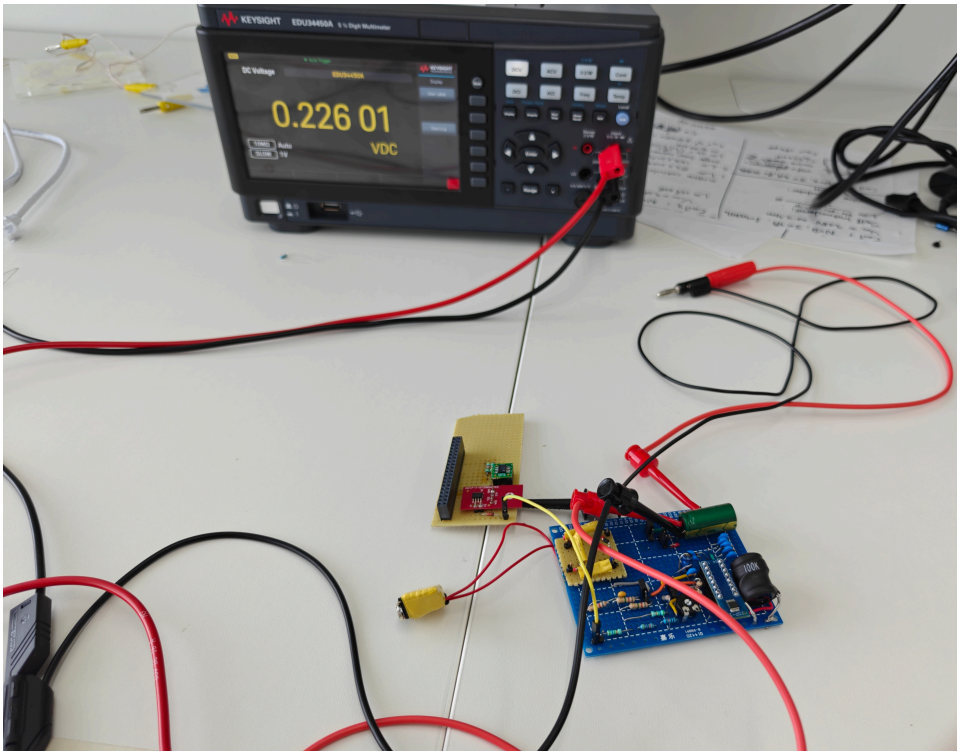

Figure A.1: Capacitor voltage measurement setup

B

Python code

B.1. Temperature readout and visualisation script

```
import os
import struct
import pandas as pd
import matplotlib

# Use TkAgg backend for embedding in Tkinter
matplotlib.use('TkAgg')

import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

import tkinter as tk
from tkinter import filedialog, scrolledtext, messagebox

def format_temp(t: float) -> str:
    s = f"{t:.4f}".rstrip('0').rstrip('.')
    return s

def select_and_process_file():
    # === Step A: Let the user pick one .bin file ===
    filepath = filedialog.askopenfilename(
        title="Select a .bin data file",
        filetypes=[("Binary files", "*.bin"), ("All files", "*.*")]
    )
    if not filepath:
        return  # user cancelled

    try:
        # === Step B: Read raw binary and unpack into 16-bit words ===
        with open(filepath, 'rb') as f:
            raw_data = f.read()
    except Exception as e:
        messagebox.showerror("File Read Error", f"Could not open file:\n{e}")
        return

    if len(raw_data) % 2 != 0:
        messagebox.showwarning(
            "Data Warning",
            "File length is not a multiple of 2. The last byte will be ignored."
        )
        raw_data = raw_data[: (len(raw_data) // 2) * 2]

    entry_count = len(raw_data) // 2
    # Unpack as big-endian unsigned shorts (2 bytes each)
    words = struct.unpack('>' + 'H' * entry_count, raw_data)

    # === Step 'B: Decode each 12-bit Q4 TMP75B reading to Celsius ===
```

```
    temperatures = []
    for raw_word in words:
        # 1) Mask off any stray high bits, keep lower 12 bits:
        twelve_bits = raw_word & 0x0FFF

        # 2) Shift left 4 to reconstruct 'TMP75Bs left-justified Q4 format:
        left_justified = twelve_bits << 4

        # 3) Sign-extend from 12 bits (now in bits 15:4 of a 16-bit value):
        if left_justified & 0x8000:
            signed16 = left_justified - 0x10000
        else:
            signed16 = left_justified

        # 4) Drop the 4 fractional bits (Q4) by shifting right 4:
        signed12 = signed16 >> 4

        # 5) Each LSB = 0.0625 °C (per TMP75B datasheet):
        temp_c = signed12 * 0.0625

        temperatures.append(temp_c)

    # === Step C: Build DataFrame for Day / HourInDay / continuous Hour index ===
    df = pd.DataFrame({
        'Temperature (°C)': temperatures,
        'Day':         [(i // 24) + 1 for i in range(entry_count)],
        'HourInDay':   [(i % 24) + 1 for i in range(entry_count)],
        'Hour':        list(range(1, entry_count + 1))
    })

    # === Step D: Display numeric table in the scrollable text widget ===
    text_widget.configure(state='normal')
    text_widget.delete("1.0", tk.END)  # clear previous contents

    current_day = 0
    for idx, row in df.iterrows():
        day = int(row['Day'])
        hour_in_day = int(row['HourInDay'])
        temp = row['Temperature (°C)']
        if day != current_day:
            current_day = day
            text_widget.insert(tk.END, f"\n--- Day {day} ---\n")
        text_widget.insert(
            tk.END,
            f"Hour {hour_in_day:02d}: {format_temp(temp)} °C\n"
        )

    text_widget.configure(state='disabled')  # make it □readonly

    # === Step E: Plot in the embedded Matplotlib canvas ===
    ax.clear()
    ax.plot(df['Hour'], df['Temperature (°C)'], linestyle='-', color='blue')

    if entry_count < 24:
        # Fewer than 24 readings: show each hour index on x-axis
        ax.set_xticks(df['Hour'])
        ax.set_xticklabels([str(h) for h in df['Hour']])
        ax.set_xlabel('Hour')
    else:
        # 24 or more: show days on x-axis
        max_day = df['Day'].max()
        day_ticks  = [(d - 1) * 24 + 1 for d in range(1, max_day + 1)]
        day_labels = [str(d) for d in range(1, max_day + 1)]
        ax.set_xticks(day_ticks)
        ax.set_xticklabels(day_labels)
        ax.set_xlabel('Day')

    ax.set_title('Temperature Over Time (Hourly Readings)')
    ax.set_ylabel('Temperature (°C)')
    ax.grid(True)

```

```
    canvas.draw()

# Close any Matplotlib figures and destroy the Tk window to ensure the process exits.
def on_closing():
    try:
        plt.close('all')
    except:
        pass
    root.destroy()


# === Main program setup ===
root = tk.Tk()
root.title("Temperature Data Logger Reader")

# Register the close-handler so the process fully exits on window close
root.protocol("WM_DELETE_WINDOW", on_closing)

# Menu bar (optional)
menubar = tk.Menu(root)
filemenu = tk.Menu(menubar, tearoff=0)
filemenu.add_command(label="Open .bin …File", command=select_and_process_file)
filemenu.add_separator()
filemenu.add_command(label="Exit", command=on_closing)
menubar.add_cascade(label="File", menu=filemenu)
root.config(menu=menubar)

# Frame for text output (left side)
text_frame = tk.LabelFrame(root, text="Numeric Readings", padx=5, pady=5)
text_frame.grid(row=0, column=0, sticky="nsew", padx=5, pady=5)

# Scrolled text widget for showing day/hour temperature lines
text_widget = scrolledtext.ScrolledText(
    text_frame,
    wrap=tk.WORD,
    width=30,
    height=20,
    state='disabled'
)
text_widget.pack(fill=tk.BOTH, expand=True)

# Frame for the plot (right side)
plot_frame = tk.LabelFrame(root, text="Temperature Plot", padx=5, pady=5)
plot_frame.grid(row=0, column=1, sticky="nsew", padx=5, pady=5)

# Embed a Matplotlib Figure + Canvas
fig, ax = plt.subplots(figsize=(6, 4))
canvas = FigureCanvasTkAgg(fig, master=plot_frame)
canvas.get_tk_widget().pack(fill=tk.BOTH, expand=True)

# Configure grid weighting so frames expand properly
root.grid_columnconfigure(0, weight=1)
root.grid_columnconfigure(1, weight=2)
root.grid_rowconfigure(0, weight=1)

# "…"Browse button
browse_btn = tk.Button(root, text="Browse & Load File", command=select_and_process_file)
browse_btn.grid(row=1, column=0, columnspan=2, pady=(0, 10))

root.mainloop()
```

B.2. Plotting measured capacitor charging script

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import PchipInterpolator

#Fill in measured data:
#iPhone 15 Pro:
t_ip = np.arange(0, 9)  # seconds -08
```

```
V_ip = np.array([0.003, 1.9, 2.55, 3, 3.20, 3.28, 3.31, 3.33, 3.34])
#OnePlus 13:
t_op = np.arange(0, 9)  # seconds -08
V_op = np.array([0.003, 1.71, 2.5, 2.86, 3.02, 3.19, 3.25, 3.28, 3.29])

from scipy.interpolate import UnivariateSpline

#spline curves
f_ip = UnivariateSpline(t_ip, V_ip, s=0.02)
f_op = UnivariateSpline(t_op, V_op, s=0.02)

#Time axis
t_fine = np.linspace(0,  8.0, 200)

plt.figure(figsize=(8, 5), dpi=300)
plt.plot(t_fine, f_ip(t_fine), '--', color='blue', label='iPhone 15 Pro (interpolated)')
plt.plot(t_fine, f_op(t_fine), '--', color='red', label='OnePlus 13 (interpolated)')
plt.scatter(t_ip, V_ip, marker='o', s=40, color='blue', label='iPhone 15 Pro (measured)')
plt.scatter(t_op, V_op, marker='s', s=40, color='red', label='OnePlus 13 (measured)')

plt.title('Measured charging of 3300 µF capacitor with Iphone and Oneplus phones')
plt.xlabel('Time [s]')
plt.ylabel('Capacitor voltage [V]')
plt.grid(True, linestyle='-', linewidth=0.5)
plt.legend()
plt.tight_layout()
plt.show()
```

C

Tabular Temperature Output from Python Application

C.1. Output for 8-Sample Dataset

```
--- Day 1 ---
Hour 01: 127.9375 °C
Hour 02: 125 °C
Hour 03: 25 °C
Hour 04: 0.0625 °C
Hour 05: 0 °C
Hour 06: -0.0625 °C
Hour 07: -25 °C
Hour 08: -40 °C
```

C.2. Partial output for 700-Sample Dataset

```
--- Day 1 ---
Hour 01: 12.5 °C
Hour 02: 12.5625 °C
Hour 03: 12.625 °C
Hour 04: 12.6875 °C
Hour 05: 12.75 °C
Hour 06: 12.75 °C
Hour 07: 12.8125 °C
Hour 08: 12.875 °C
Hour 09: 12.9375 °C
Hour 10: 13 °C
Hour 11: 13.0625 °C
Hour 12: 13.125 °C
Hour 13: 13.1875 °C
Hour 14: 13.25 °C
Hour 15: 13.25 °C
Hour 16: 13.3125 °C
Hour 17: 13.375 °C
Hour 18: 13.4375 °C
Hour 19: 13.5 °C
Hour 20: 13.5625 °C
Hour 21: 13.625 °C
Hour 22: 13.625 °C
Hour 23: 13.6875 °C
```

```
Hour 24: 13.75 °C

--- Day 2 ---
Hour 01: 13.8125 °C
Hour 02: 13.875 °C
Hour 03: 13.9375 °C
Hour 04: 13.9375 °C
Hour 05: 14 °C
Hour 06: 14.0625 °C
Hour 07: 14.125 °C
Hour 08: 14.125 °C
Hour 09: 14.1875 °C
Hour 10: 14.25 °C
Hour 11: 14.3125 °C
Hour 12: 14.3125 °C
Hour 13: 14.375 °C
Hour 14: 14.4375 °C
Hour 15: 14.5 °C
Hour 16: 14.5 °C
Hour 17: 14.5625 °C
Hour 18: 14.625 °C
Hour 19: 14.625 °C
Hour 20: 14.6875 °C
Hour 21: 14.75 °C
Hour 22: 14.75 °C
Hour 23: 14.8125 °C
Hour 24: 14.8125 °C

--- Day 3 ---
Hour 01: 14.875 °C
Hour 02: 14.9375 °C
Hour 03: 14.9375 °C
Hour 04: 15 °C
Hour 05: 15 °C
Hour 06: 15.0625 °C
Hour 07: 15.0625 °C
Hour 08: 15.125 °C
Hour 09: 15.125 °C
Hour 10: 15.1875 °C
Hour 11: 15.1875 °C
Hour 12: 15.25 °C
Hour 13: 15.25 °C
Hour 14: 15.25 °C
Hour 15: 15.3125 °C
Hour 16: 15.3125 °C
Hour 17: 15.375 °C
Hour 18: 15.375 °C
Hour 19: 15.375 °C
Hour 20: 15.4375 °C
Hour 21: 15.4375 °C
Hour 22: 15.4375 °C
Hour 23: 15.5 °C
Hour 24: 15.5 °C
```

Bibliography

[1] R. Badia-Melis, U. Mc Carthy, L. Ruiz-Garcia, J. Garcia-Hierro, and J. Robla Villalba, "New trends in cold chain monitoring applications - a review," *Food Control*, vol. 86, pp. 170–182, 2018, ISSN: 0956-7135. DOI: https://doi.org/10.1016/j.foodcont.2017.11.022.

[2] M. Lana, L. Tijskens, and O. Kooten, "Effects of storage temperature and fruit ripening on firmness of fresh cut tomatoes," *Postharvest Biology and Technology*, vol. 35, no. 1, pp. 87–95, 2005, ISSN: 0925-5214. DOI: https://doi.org/10.1016/j.postharvbio.2004.07.001.

[3] Cargo Handbook, *Bananas*. [Online]. Available: https://www.cargohandbook.com/Bananas.

[4] V. C. Falcón, Y. V. V. Porras, C. M. G. Altamirano, and U. Kartoglu, "A vaccine cold chain temperature monitoring study in the united mexican states," *Vaccine*, vol. 38, no. 33, pp. 5202–5211, 2020, ISSN: 0264-410X. DOI: https://doi.org/10.1016/j.vaccine.2020.06.014.

[5] United Nations University (UNU), *Global e-waste surging: Up 21% in 5 years*, 2020. [Online]. Available: https://unu.edu/press-release/global-e-waste-surging-21-5-years.

[6] U. N. I. for Training and Research, *Global e-waste monitor 2024: Electronic waste rising five times faster than documented e-waste recycling*, 2024. [Online]. Available: https://unitar.org/about/news-stories/press/global-e-waste-monitor-2024-electronic-waste-rising-five-times-faster-documented-e-waste-recycling.

[7] Sensitech, *Shipping temperature indicators*, 2025. [Online]. Available: https://www.sensitech.com/en/products/indicators/.

[8] ELPRO, *Sustainable libero cx multi-use pdf data loggers*, 2025. [Online]. Available: https://www.elpro.com/en/libero-cx-multi-use?hsCtaTracking=b48d56d4-9aca-42f8-a58c-10e01626d341%7Cbfd81ba4-70f3-48fd-bb41-f276e150d6a1.

[9] LogTag, *Trix-8 multi-use temperature logger*, 2025. [Online]. Available: https://logtagrecorders.com/product/trix-8/.

[10] N. Khalid, R. Mirzavand, and A. K. Iyer, "A survey on battery-less rfid-based wireless sensors," *Micromachines*, vol. 12, no. 7, 2021, ISSN: 2072-666X. DOI: 10.3390/mi12070819. [Online]. Available: https://www.mdpi.com/2072-666X/12/7/819.

[11] G. T. Le, T. V. Tran, H.-S. Lee, and W.-Y. Chung, "Long-range batteryless rf sensor for monitoring the freshness of packaged vegetables," *Sensors and Actuators A: Physical*, vol. 237, pp. 20–28, 2016, ISSN: 0924-4247. DOI: https://doi.org/10.1016/j.sna.2015.11.013.

[12] W.-Y. Chung, G. T. Le, T. V. Tran, and N. H. Nguyen, "Novel proximal fish freshness monitoring using batteryless smart sensor tag," *Sensors and Actuators B: Chemical*, vol. 248, pp. 910–916, 2017, ISSN: 0925-4005. DOI: https://doi.org/10.1016/j.snb.2017.01.134.

[13] G. Loubet, A. Takacs, E. Gardner, A. De Luca, F. Udrea, and D. Dragomirescu, "Lorawan battery-free wireless sensors network designed for structural health monitoring in the construction domain," *Sensors*, vol. 19, no. 7, 2019, ISSN: 1424-8220. DOI: 10.3390/s19071510. [Online]. Available: https://www.mdpi.com/1424-8220/19/7/1510.

[14] A. Sidibe, G. Loubet, A. Takacs, and D. Dragomirescu, "A multifunctional battery-free bluetooth low energy wireless sensor node remotely powered by electromagnetic wireless power transfer in far-field," *Sensors*, vol. 22, no. 11, 2022, ISSN: 1424-8220. DOI: 10.3390/s22114054. [Online]. Available: https://www.mdpi.com/1424-8220/22/11/4054.

[15] R. Dekimpe, P. Xu, M. Schramme, P. Gérard, D. Flandre, and D. Bol, "A battery-less ble smart sensor for room occupancy tracking supplied by 2.45-ghz wireless power transfer," *Integration*, vol. 67, pp. 8–18, 2019, ISSN: 0167-9260. DOI: https://doi.org/10.1016/j.vlsi.2019.03.006.

[16] S. H. Shah and I. Yaqoob, “A survey: Internet of things (iot) technologies, applications and challenges,” in *2016 IEEE Smart Energy Grid Engineering (SEGE)*, 2016, pp. 381–385. DOI: 10.1109/SEGE.2016.7589556.

[17] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, “Internet of things (iot) communication protocols: Review,” in *2017 8th International Conference on Information Technology (ICIT)*, 2017, pp. 685–690. DOI: 10.1109/ICITECH.2017.8079928.

[18] G. Paolini, M. Guermandi, D. Masotti, *et al.*, “Rf-powered low-energy sensor nodes for predictive maintenance in electromagnetically harsh industrial environments,” *Sensors*, vol. 21, no. 2, 2021, ISSN: 1424-8220. DOI: 10.3390/s21020386. [Online]. Available: https://www.mdpi.com/1424-8220/21/2/386.

[19] D. K. Singh and R. Sobti, “Wireless communication technologies for internet of things and precision agriculture: A review,” in *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC)*, 2021, pp. 765–769. DOI: 10.1109/ISPCC53510.2021.9609421.

[20] S. Bhatia, Z. A. Jaffery, and S. Mehfuz, “A comparative study of wireless communication protocols for use in smart farming framework development,” in *2023 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, 2023, pp. 1–7. DOI: 10.1109/ICCT56969.2023.10075696.

[21] G. Loubet, A. Takacs, E. Gardner, A. De Luca, F. Udrea, and D. Dragomirescu, “Lorawan battery-free wireless sensors network designed for structural health monitoring in the construction domain,” *Sensors*, vol. 19, no. 7, 2019, ISSN: 1424-8220. DOI: 10.3390/s19071510. [Online]. Available: https://www.mdpi.com/1424-8220/19/7/1510.

[22] L. Alliance, *Lorawan specification v1.1*, 2025. [Online]. Available: https://resources.lora-alliance.org/technical-specifications/lorawan-specification-v1-1.

[23] STMicroelectronics, “ST25DV04KC / ST25DV16KC / ST25DV64KC – Dynamic NFC/RFID Tag IC with 4-, 16-, or 64-Kbit EEPROM, Fast-Transfer-Mode Capability, and Optimized I^{2}C,” STMicroelectronics, Tech. Rep. DS13519 Rev 8, Jul. 2024. [Online]. Available: https://www.st.com/resource/en/datasheet/st25dv64kc.pdf.

[24] onsemi, “N24RF64 – Dual-Interface ISO 15693 RFID 64-Kb EEPROM Tag with I^{2}C Interface,” onsemi, Tech. Rep. Rev 2, May 2025. [Online]. Available: https://www.onsemi.com/download/data-sheet/pdf/n24rf64-d.pdf.

[25] STMicroelectronics, “M24SR64-Y – Dynamic NFC/RFID Tag IC with 64-Kbit EEPROM, NFC Forum Type 4 Tag and I^{2}C Interface,” STMicroelectronics, Tech. Rep. DocID023790 Rev 20, Aug. 2017. [Online]. Available: https://www.st.com/resource/en/datasheet/m24sr64-y.pdf.

[26] NXP Semiconductors, “NTAG 213/215/216 – NFC Forum Type 2 Tag Compliant ICs with 144 / 504 / 888 Bytes User Memory,” NXP Semiconductors, Tech. Rep. Product Data Sheet Rev 3.2, Jun. 2015. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/NTAG213_215_216.pdf.

[27] NXP Semiconductors, “NTAG 424 DNA and NTAG 424 DNA TagTamper – Secure NFC T4T compliant ICs,” NXP Semiconductors, Tech. Rep. Rev. 3.0, Jan. 2019. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/NT4H2421Gx.pdf.

[28] STMicroelectronics, “Energy harvesting delivery impact on st25dv-i2c series behaviour during rf communication,” STMicroelectronics, Application Note AN4913 Rev. 2, Jun. 2021. [Online]. Available: https://www.st.com/resource/en/application_note/an4913-energy-harvesting-delivery-impact-on-st25dvi2c-series-behaviour-during-rf-communication-stmicroelectronics.pdf.

[29] onsemi, *N24RF64E – rfid 64 kb eeprom tag, iso 15693 rf, i^{2}c bus, energy harvesting*, Datasheet, ON Semiconductor, Rev. 1, Nov. 2018. [Online]. Available: https://www.onsemi.com/download/data-sheet/pdf/n24rf64e-d.pdf.

[30] Farnell, *“st25dv64kc” dynamic nfc/rfid tag ic product listing*, https://nl.farnell.com/search?st=ST25DV64KC, Accessed 12 Jun 2025.

[31] R. Components, *"m24sr64-y" nfc tag ic product listing*, https://nl.rs-online.com/web/c/?searchType=MPN&searchTerm=M24SR64-Y, Accessed 12 Jun 2025.

[32] Farnell, *"n24rf64" rfid ic product listing*, https://nl.farnell.com/c/semiconductors-ics/rf/rfid?st=N24RF64, Accessed 12 Jun 2025.

[33] STMicroelectronics, "Ndef management with st25dv-i2c series, and st25tv16k and st25tv64k products," STMicroelectronics, Application note AN4911, May 2021. [Online]. Available: https://www.st.com/resource/en/application_note/an4911-ndef-management-with-st25dvi2c-series-and-st25tv16k-and-st25tv64k-products-stmicroelectronics.pdf.

[34] STMicroelectronics, "How to design an antenna for dynamic nfc tags," STMicroelectronics, Application Note AN2972, 2019. [Online]. Available: https://www.st.com/resource/en/application_note/an2972-how-to-design-an-antenna-for-dynamic-nfc-tags-stmicroelectronics.pdf.

[35] STMicroelectronics, "Design of a 13.56 mhz double-layer antenna," STMicroelectronics, Application Note AN5605, Feb. 2021. [Online]. Available: https://www.st.com/resource/en/application_note/an5605-design-of-a-1356-mhz-doublelayer-antenna-stmicroelectronics.pdf.

[36] STMicroelectronics, *Nfc inductance*, 2025. [Online]. Available: https://eds.st.com/antenna/#/.

[37] STMicroelectronics, "14 mm x 14 mm antenna reference board for the st25dv64kc dual interface eeprom," STMicroelectronics, Data brief DB4519, Apr. 2022. [Online]. Available: https://www.st.com/resource/en/data_brief/ant7-t-25dv64kc.pdf.

[38] Vishay Semiconductors, *1N4148: Small signal fast switching diode datasheet*, 81857, Rev. 1.6, Document No. 81857, Vishay Intertechnology, Inc., Malvern, PA, USA, 2024. [Online]. Available: https://www.vishay.com/docs/81857/1n4148.pdf.

[39] STMicroelectronics, *Application executable of the nfc tap android app for st25 product family*, 2025. [Online]. Available: https://www.st.com/en/embedded-software/stsw-st25001.html.

[40] Texas Instruments, "TMP75B 1.8-V Digital Temperature Sensor with Two-Wire Interface and Alert," Texas Instrumnets, Tech. Rep., Aug. 2014. [Online]. Available: https://www.ti.com/lit/ds/symlink/tmp75b.pdf?ts=1749820766458&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTMP75B.

[41] Texas Instruments, "How to Read and Interpret Digital Temperature Sensor Output Data," Texas Instrumnets, Application Note, Jan. 2025. [Online]. Available: https://www.ti.com/lit/an/sbaa588a/sbaa588a.pdf?ts=1749799556583.

[42] Keysight Technologies, "EDU34450A 5½-Digit Dual□Display Digital Multimeter: Data Sheet," Keysight Technologies, Product Datasheet 3121-1002EN, 2022. [Online]. Available: https://www.keysight.com/us/en/assets/3121-1002/data-sheets/EDU34450A-5-5-Digit-Dual-Display-Digital-Multimeter.pdf (visited on 06/15/2025).