TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Introduction to STM32 Blue Pill Microcontroller

In recent years, microcontrollers have become integral to modern electronics, powering everything from household appliances to advanced industrial systems. Among the wide range of microcontrollers available, the STM32F103C8T6, commonly referred to as the "Blue Pill," has garnered considerable attention from hobbyists, engineers, and developers due to its low cost, high performance, and rich peripheral set. The project presented in this report focuses on understanding, exploring, and implementing the STM32F103C8T6 microcontroller in embedded system applications. The project serves as both an academic endeavor and a practical exercise in embedded system design, targeting learners and professionals seeking cost-effective yet powerful development platforms.

The STM32F103C8T6 Blue Pill is based on the ARM Cortex-M3 core, operating at 72 MHz, and is equipped with 64 KB of Flash memory and 20 KB of SRAM. It supports various communication protocols such as USART, I2C, SPI, and USB, along with multiple timers, ADC channels, and GPIOs. These features make it highly suitable for embedded systems development, especially in applications involving sensor data acquisition, motor control, and real-time signal processing. The availability of numerous open-source tools like STM32CubeIDE, Keil u Vision, and Platform IO has made it easier than ever to program and debug the Blue Pill board, lowering the barrier to entry for students and developers.

This project aims to build a comprehensive understanding of the Blue Pill's hardware and software features by developing a series of test programs and real-world applications. The intent is to demonstrate how such a low-cost module can serve as the heart of a sophisticated embedded solution. In this context, the STM32F103C8T6 becomes not just a study subject but a platform for experimentation and innovation. It allows developers to design and test prototypes that can later be scaled into commercial products. This hands-on approach is vital for students of Electronics and Communication Engineering, providing a clear link between theoretical knowledge and its practical implementation.

The Blue Pill has also become a prominent alternative to more expensive development boards like those from the Arduino or official STMicroelectronics series. Although STMicroelectronics does not officially support the Blue Pill as a reference development board, its popularity has surged due to its availability in the maker community and compatibility with STM's official tools. Furthermore, its ability to be programmed via ST-Link, USB-to-Serial converters, or even using bootloader-based methods adds to its

versatility in educational and prototyping environments. By working with this microcontroller, students gain a deeper towards intelligent automation, IoT, and real-time systems, the ability to work with microcontrollers becomes crucial. Through this project, learners are introduced to the foundational elements of embedded systems: memory management, GPIO configuration, ADC/DAC operations, PWM control, and serial communication. By systematically exploring these elements, the report ensures that readers gain hands-on familiarity with microcontroller programming, while simultaneously understanding how such components can be integrated into larger systems. The use of real code examples, circuit diagrams, and implementation challenges provides a holistic learning experience.

In summary, this project revolves around the STM32F103C8T6 Blue Pill microcontroller and aims to provide a structured exploration of its features, usage, and real-time applications. It is designed to bridge the gap between academic concepts and industry requirements, enabling learners to become proficient in embedded system development. The microcontroller's affordability and robustness make it ideal for use in academic research, DIY projects, and commercial prototypes alike. Through systematic study and practical experimentation, this report will demonstrate the Blue Pill's potential and prepare students for real-world engineering challenges.

The STM32F103C8T6, commonly referred to as the "Blue Pill" board, is a powerful yet affordable development board widely used by embedded engineers, hobbyists, and students for rapid prototyping and implementation of real-time embedded systems. Based on the ARM Cortex-M3 core, it is clocked at 72 MHz, making it suitable for a wide range of applications that require real-time response, low power consumption, and efficient peripheral integration. This microcontroller supports a variety of input/output interfaces, including UART, SPI, I2C, and USB, which provide designers with the flexibility to integrate sensors, actuators, communication modules, and other peripherals seamlessly.

The goal of this project is to explore and understand the interfacing capabilities and core functionality of the STM32F103C8T6 microcontroller by implementing a real-world use case. Through this project, we aim to demonstrate how the STM32 Blue Pill board can be effectively utilized for embedded system development, particularly in applications such as sensor data acquisition, communication over standard protocols, control system implementation, and low-power operations.

1.2 Motivation

The proliferation of smart systems and the Internet of Things (IoT) has transformed how data is collected, processed, and transmitted across diverse industries—from agriculture and logistics to environmental monitoring and infrastructure management. In these domains, the need for reliable, long-distance, and energy-efficient communication protocols is more critical than ever. Existing short-range technologies like Wi-Fi and Bluetooth suffer from significant limitations when applied in vast geographical areas or power-constrained environments.

LoRa (Long Range) technology, particularly the SX1276 transceiver module, presents a compelling solution to these challenges. Its ability to maintain low-power, long-range wireless communication over unlicensed ISM bands makes it ideal for distributed sensor networks and remote embedded systems. The inherent characteristics of LoRa—such as chirp spread spectrum modulation, robust data transmission, and support for star network topologies—offer significant advantages in terms of scalability, penetration, and resilience in challenging conditions.

Simultaneously, the STM32F103C8T6 Blue Pill microcontroller offers a robust yet cost-effective platform for embedded system design. Built on the ARM Cortex-M3 core, it provides ample computational power, multiple I/O interfaces, and extensive community support. This makes it a preferred choice for prototyping and deploying real-time applications that demand precise control, modularity, and low power consumption.

The integration of these two technologies—LoRa SX1276 and STM32 Blue Pill—serves as a foundation for creating low-power wide-area network (LPWAN) applications that are not only technically efficient but also economically viable. The motivation behind this project stems from a desire to harness the strengths of these components in real-world scenarios where traditional wireless solutions fall short.

Furthermore, this initiative is aligned with the broader goals of sustainability and digital transformation. It supports the development of smart systems that reduce manual monitoring, conserve energy, and extend connectivity to underserved or geographically isolated regions. From a research and academic standpoint, the project also provides valuable insights into embedded communication systems, offering a platform for further innovation in wireless technology and IoT infrastructure.

Traditional wireless communication technologies like Wi-Fi, Zigbee, and Bluetooth are limited in terms of range and power efficiency. These limitations hinder their use in large-scale deployments or in locations without infrastructure support. In contrast, LoRa

(Long Range) technology, specifically the SX1276 transceiver module, is engineered to overcome these constraints. LoRa's unique chirp spread spectrum modulation enables long-range communication (several kilometres in open environments) while maintaining low power consumption, making it a suitable candidate for battery-powered or solar-powered devices.

Parallel to this, the STM32F103C8T6 Blue Pill microcontroller offers a low-cost yet powerful computing platform with essential features such as multiple communication protocols (SPI, UART, I2C), GPIO flexibility, and fast processing with ARM Cortex-M3 architecture. It is widely adopted in embedded systems due to its accessibility, affordability, and support from open-source communities and development tools.

The integration of LoRa SX1276 with the STM32 Blue Pill is not only technically compelling but also deeply aligned with global efforts to build energy-efficient and sustainable smart systems. This pairing allows for the development of a decentralized network of intelligent nodes capable of collecting and transmitting data over wide areas with minimal maintenance. Such systems are critical for improving efficiency in agriculture, environmental monitoring, industrial automation, and public infrastructure.

From an academic and developmental standpoint, this project motivates exploration into advanced embedded communication techniques, firmware development, low-power optimization, and system integration. It opens doors for future enhancements such as cloud-based data logging, machine learning-based sensor analysis, and integration into larger IoT ecosystems.

Ultimately, the motivation for this project stems from a desire to bridge the gap between theoretical knowledge and real-world applications, while contributing to the creation of scalable, cost-effective, and sustainable wireless communication systems that address current and emerging global challenges.

1.3 Objectives of the Project

The primary objective of this project is to explore and implement a reliable and efficient long-range communication system by interfacing the LoRa SX1276 transceiver module with the STM32F103C8T6 Blue Pill microcontroller. The convergence of these two technologies is aimed at addressing the growing demand for robust, low-power, and wide-area wireless communication, which is especially important for the development of modern Internet of Things (IoT) systems. This project is motivated by the need to provide a communication interface that can overcome the limitations of conventional technologies like Wi-Fi and Bluetooth in terms of both range and energy efficiency, making it suitable for deployment in remote, outdoor, or infrastructure-less environments.

One of the fundamental goals of this project is to develop a strong theoretical and practical understanding of the STM32F103C8T6 microcontroller. This involves studying its internal architecture, particularly its ARM Cortex-M3 core, and learning how to utilize its peripheral interfaces such as SPI (Serial Peripheral Interface), UART, I2C, GPIOs, and timers. Familiarity with these hardware resources is essential for implementing a reliable interface between the microcontroller and external modules such as the LoRa SX1276. To accomplish this, the project includes setting up a development environment using STM32CubeIDE, configuring the toolchain for embedded C programming, and writing low-level code to initialize and manage the hardware components.

Another critical objective is to investigate the features and operational characteristics of the LoRa SX1276 module. This involves understanding its radio communication protocol, modulation technique (chirp spread spectrum), data rate control, bandwidth settings, transmission power, and receiver sensitivity. The module's support for ISM band operation and its flexible configuration options make it highly suitable for long-distance data exchange. A key part of the project is to integrate this module with the STM32 microcontroller using SPI communication and develop firmware that enables configuration, packet transmission, and reception.

Once the interface between the STM32 and the LoRa module is established, the next objective is to implement a functional point-to-point (or basic mesh) communication system. This includes writing embedded software that allows one node to send data and another node to receive it reliably over a distance. To validate this system, the project involves conducting real-world experiments to test the signal strength, packet delivery ratio, latency, and maximum achievable communication range under different environmental conditions. These performance metrics will be analyzed to determine the practical limitations and potential applications of the developed system.

The project also emphasizes the importance of low-power operation, which is crucial for battery-operated or solar-powered systems deployed in remote locations. Therefore, another major objective is to measure and optimize the energy consumption of the entire setup. This may involve exploring the STM32's power-saving modes and the LoRa module's sleep configurations to ensure energy efficiency during idle periods, without compromising communication reliability.

Furthermore, this project aims to lay the foundation for future scalability and integration with real-world IoT use cases. This includes designing the system in a modular way that allows easy expansion by integrating sensors such as temperature, humidity, or motion detectors. In doing so, the project provides a base that can evolve into more complex systems like smart farming networks, environmental monitoring stations, or asset tracking solutions. It also sets the stage for future enhancements like gateway-based network topologies, cloud connectivity via MQTT/HTTP protocols, and advanced data processing through edge computing.

In addition to the technical goals, this project serves an educational purpose by offering hands-on experience in embedded systems design, wireless communication protocols, microcontroller programming, and hardware-software integration. It is expected that the knowledge and skills developed through this work will contribute to the academic growth of the student, while also providing practical solutions to communication challenges in the real world.

In summary, the objectives of this project are multifaceted: they involve technical exploration of microcontrollers and radio transceivers, practical implementation of embedded communication systems, performance validation under various conditions, energy efficiency optimization, and preparing the platform for scalable IoT applications. These objectives collectively contribute to the broader vision of enabling smart, connected, and sustainable systems that can function reliably over long distances with minimal infrastructure and power requirements.

1.4 Product Vision Statement: STM32F103C8T6 for Embedded Systems

The STM32F103C8T6 microcontroller, commonly referred to as the "Blue Pill," represents a powerful yet cost-effective platform for modern embedded system development. Its product vision is rooted in delivering accessible, efficient, and scalable microcontroller solutions that empower engineers, developers, and researchers to prototype and deploy real-world applications with confidence and flexibility. In an era where embedded intelligence and edge computing are becoming indispensable across industries, the STM32F103C8T6 positions itself as a versatile enabler of innovation through its robust architecture, peripheral integration, and performance-to-cost ratio.

Built around the ARM Cortex-M3 core, the STM32F103C8T6 combines 32-bit processing performance with low power consumption, real-time responsiveness, and comprehensive peripheral support. These features make it highly suitable for a broad spectrum of embedded applications, including industrial control systems, robotics, wireless communication, automation, and IoT (Internet of Things). The product vision is not merely about offering processing power; rather, it emphasizes creating a flexible platform that can interface effortlessly with sensors, actuators, communication modules, and other peripherals—paving the way for intelligent and autonomous systems.

At the heart of this vision is accessibility. Unlike many proprietary embedded solutions, the STM32F103C8T6 is supported by a rich open-source ecosystem and affordable development tools, such as STM32CubeIDE, PlatformIO, and Arduino-compatible environments. This ensures that developers at all levels—from hobbyists to professionals—can rapidly design, test, and iterate on embedded projects without facing high entry barriers. The microcontroller's compatibility with common programming frameworks, libraries, and communication protocols allows seamless integration with other modules, such as the LoRa SX1276 used in this project, thus reinforcing its role as a central processing unit in wireless sensor networks and embedded systems.

Another core component of the STM32F103C8T6 product vision is sustainability and long-term deployment readiness. Its low-power modes, watchdog timers, and precise control over hardware resources allow developers to build energy-efficient systems suited for remote or battery-operated environments. This aligns with the growing emphasis on sustainable technology, where devices are expected to operate for extended periods with minimal maintenance. Whether used in agricultural monitoring, smart cities, or healthcare devices, the STM32F103C8T6 supports design strategies that are both cost-effective and environmentally conscious.

Moreover, the STM32 family, including the F1 series, benefits from long-term availability and consistent updates from STMicroelectronics, ensuring developers can rely on these microcontrollers for commercial-grade products. This commitment enhances the vision of the STM32F103C8T6 as more than just a development board—it is a gateway to deploying real-world solutions that require reliability, longevity, and community-driven support.

In the evolving landscape of embedded systems, the demand for powerful, cost-effective, and adaptable microcontroller platforms is more significant than ever. The STM32F103C8T6 Blue Pill microcontroller represents a well-balanced solution, bringing together processing capability, low power consumption, and peripheral richness in a compact and affordable form. The product vision for this microcontroller

is centered on empowering innovation in embedded design—whether in academic research, industrial automation, consumer electronics, or smart infrastructure.

At its core, the STM32F103C8T6 is built on the ARM Cortex-M3 processor, known for its high efficiency, low interrupt latency, and 32-bit processing capability, making it ideal for time-critical embedded tasks. The vision driving this microcontroller's integration into embedded systems is one of democratized development—providing a reliable and feature-rich platform to hobbyists, students, startups, and engineers at a fraction of the cost of traditional proprietary solutions. It facilitates easy learning and rapid prototyping without compromising on advanced features such as DMA controllers, multiple communication interfaces (SPI, I2C, UART, CAN), PWM timers, and ADCs.

The Blue Pill's popularity in the maker and academic community has further shaped its vision: to be a gateway to practical learning and real-world application. It offers a hands-on platform where developers can explore microcontroller programming, embedded C/C++ development, real-time control systems, and IoT implementation. The board's support by open-source toolchains such as STM32CubeIDE, Keil, PlatformIO, and even Arduino IDE, combined with a vast community and documentation, fosters an inclusive environment for development. This accessibility ensures that the microcontroller can serve as both a learning tool and a foundation for sophisticated embedded products

Another key dimension of this product vision is its focus on modularity and flexibility. The STM32F103C8T6 is well-suited for integration with various sensors, actuators, and communication modules—including LoRa, Wi-Fi, GSM, and BLE—making it a natural fit for scalable and interconnected embedded solutions. Its widespread use in DIY electronics, robotics, sensor networks, and automation systems highlights its adaptability across diverse domains.

1.5 Product Goals

The overarching goal of the STM32F103C8T6-based embedded system project is to develop a robust, scalable, and energy-efficient communication platform suitable for various real-world applications, particularly in remote sensing and wireless data transmission. At its core, the project aims to harness the strengths of the STM32F103C8T6 microcontroller—its 32-bit ARM Cortex-M3 processor, integrated peripherals, and low power operation—to build a reliable foundation for wireless embedded solutions. This includes not only establishing seamless SPI communication between the microcontroller and the LoRa SX1276 module but also ensuring that the

entire system can operate effectively across long distances with minimal energy consumption.

A key product goal is modularity and reusability. The system should be designed in such a way that it can easily be adapted or extended to suit other embedded applications, such as environmental monitoring, industrial automation, smart agriculture, or remote diagnostics. This means using a clean, well-structured firmware architecture that can be easily maintained and enhanced in the future. Additionally, the hardware setup should support the integration of various sensor types, power management systems, and communication modules, providing a flexible development platform.

Another important goal is to deliver a system that is both technically sound and economically viable. Given that one of the main attractions of the STM32F103C8T6 is its low cost, the project aims to demonstrate that sophisticated embedded solutions can be built on affordable hardware without sacrificing performance or reliability. This aligns with the broader mission of promoting accessible technology development for students, engineers, and communities with limited resources.

The project also prioritizes real-time data transmission and system responsiveness. This involves optimizing communication latency, data integrity, and reliability under varying conditions, including urban environments, open fields, and areas with potential signal interference. The firmware must be capable of error detection and retransmission mechanisms to ensure robust performance.

Another critical goal is to explore and implement power-efficient strategies, such as utilizing the STM32's low-power modes and configuring the LoRa module's sleep and standby settings. These features are essential for applications that rely on battery or solar power and are deployed in locations where frequent maintenance is not feasible. Reducing energy consumption while maintaining communication quality is a cornerstone of the project’s objectives.

Additionally, the system should include diagnostic and monitoring capabilities, allowing users to verify the status of transmission, signal strength, and battery life (if applicable). This feature adds value by enabling proactive system management and ensuring long-term operational reliability in the field.

1.6 Literature Review:

The STM32F103C8T6, also popularly known as the Blue Pill, has attracted widespread attention in the academic and industrial communities due to its exceptional balance of performance, cost, and functionality. This literature review aims to explore existing studies, technical documentation, and practical projects that contribute to a better

understanding of this microcontroller's role in embedded system development. By evaluating relevant sources, we can identify the motivations, capabilities, and design methodologies associated with using the STM32F103C8T6 in various applications.

Several technical papers and online articles highlight the evolution of 32-bit microcontrollers and the significant shift from 8-bit and 16-bit architectures to ARM Cortex-M-based systems, such as those in the STM32 series. The STM32F103C8T6, built around the Cortex-M3 core, has become a benchmark for developers seeking real-time processing power without compromising affordability. Its RISC-based architecture, low power consumption, and rich peripheral set—including timers, ADCs, SPI, I2C, UART, CAN, and USB—make it a preferred choice in domains like automation, control systems, and wireless communication.

Academic research has emphasized the use of the Blue Pill in low-power IoT systems, where its sleep modes and interrupt-driven features enable efficient energy management. Studies on environmental monitoring systems and agricultural applications have demonstrated that the STM32F103C8T6 can be integrated with wireless technologies like LoRa, Zigbee, and Bluetooth to collect and transmit sensor data from remote locations. These implementations often use open-source firmware frameworks such as STM32CubeMX and PlatformIO to streamline peripheral configuration and code generation.

The microcontroller's compatibility with LoRa communication protocols, as in the current project, is particularly well-documented in recent literature. Researchers have leveraged this pairing to establish reliable long-range wireless sensor networks that function in harsh or inaccessible environments. LoRa's ability to transmit data over several kilometers, combined with the STM32's precise control over communication timing and low power features, has proven effective in telemetry systems, industrial monitoring, and smart agriculture.

Literature also reveals recurring themes about educational use of the STM32F103C8T6. Many engineering institutions incorporate it into their embedded systems or microcontroller courses due to its affordability, practical, and the breadth of learning opportunities it provides—from register- features level programming to real-time debugging and peripheral interfacing. Its use in final-year student projects has become commonplace, particularly for applications involving automation, home security systems, wireless communication, and biomedical data monitoring.

In the context of embedded system design, the STM32F103C8T6 is often compared with more advanced STM32 models like the F4 and F7 series. While the latter offer enhanced computational abilities and memory, the Blue Pill continues to be chosen for cost-sensitive applications that do not require floating-point units or high-speed DSP.

This highlights its unique positioning as a mid-tier MCU ideal for real-time, low-complexity tasks that demand reliability and energy efficiency.

In conclusion, the literature reveals a consistent appreciation for the STM32F103C8T6 as a versatile and dependable microcontroller platform. Its adoption spans across academic, commercial, and hobbyist domains, each leveraging its processing capabilities, peripheral richness, and community support. This broad spectrum of applications and research findings establishes the STM32F103C8T6 as a cornerstone of embedded development, especially when paired with emerging technologies like LoRa for efficient wireless communication. The insights drawn from these studies provide a strong foundation for the current project, guiding both the technical choices and implementation strategies.

Several studies have highlighted the role of microcontrollers in condition monitoring systems, especially those utilizing sensor-based data collection such as vibration analysis. For instance, [Author, Year] demonstrated how accelerometer data processed through microcontrollers could detect early signs of machine wear. These systems often benefit from the STM32's integrated ADCs, DMA support, and real-time data processing capabilities.

The integration of wireless modules like LoRa SX1276 with STM32 microcontrollers, as discussed in technical articles and tutorials (e.g., How2Electronics, 2023), has expanded the application of these systems into remote monitoring. The long-range and low-power communication offered by LoRa makes it a viable solution for industrial environments where wired connections are impractical.

CHAPTER 2: LITERATURE SURVEY

The rapid progression of embedded system technology has revolutionized the way modern electronic systems are designed and implemented. Embedded systems, by definition, are dedicated computing systems designed to perform specific tasks within larger mechanical or electrical systems. These systems are integral to the development of smart technologies, ranging from consumer electronics and automotive systems to industrial automation and medical devices. As the demand for intelligent and energy-efficient systems continues to grow, microcontroller units (MCUs) have emerged as the fundamental hardware blocks enabling the integration of sensing, actuation, computation, and communication in embedded platforms.

One of the most significant developments in embedded hardware design has been the transition from simple, task-specific processors to sophisticated, feature-rich

microcontrollers capable of multitasking, low-power operation, and high-speed data processing. This transition has been catalyzed by the advancement of microcontroller architectures, with manufacturers such as Microchip, Texas Instruments, NXP, and STMicroelectronics at the forefront of innovation. Among these, STMicroelectronics has garnered widespread recognition for its STM32 microcontroller family, which has become one of the most widely adopted platforms in both academic research and industrial applications.

The STM32 family, based on ARM Cortex-M cores, is notable for its scalability, modularity, and extensive peripheral support. First introduced in 2007, the STM32 series has evolved into multiple sub-families, including STM32F, STM32L, STM32G, STM32H, and STM32WB/WL, each targeting specific application domains such as high performance, ultra-low-power operation, and integrated wireless communication. These microcontrollers are supported by a comprehensive ecosystem, including development tools (STM32CubeMX, STM32CubeIDE), firmware libraries (HAL/LL), and a robust user community. The widespread use of STM32 devices is a testament to their reliability, cost-effectiveness, and design flexibility.

Among the many variants within the STM32 family, the STM32F103C8T6—popularly known as the "Blue Pill"—stands out for its popularity among hobbyists, educators, and developers. The Blue Pill board is a low-cost development board that features the STM32F103C8T6 MCU, a 32-bit ARM Cortex-M3 processor with 64 KB of flash memory and 20 KB of SRAM. Despite its affordability, the board includes a broad range of peripherals such as GPIOs, ADCs, timers, PWM, I2C, SPI, USART, and CAN interfaces, making it ideal for prototyping a wide array of applications.

2.1 Evolution of STM32 Microcontroller Series:

The STM32 family of microcontrollers, developed by STMicroelectronics, represents a significant milestone in the embedded systems landscape. Since their introduction in 2007, STM32 microcontrollers have become a cornerstone of both commercial and research-oriented embedded applications. Built upon the highly efficient ARM Cortex-M processor cores, STM32 MCUs are widely recognized for their scalability, power efficiency, and rich feature sets. The evolution of the STM32 microcontroller series reflects STMicroelectronics' commitment to providing a comprehensive and flexible platform capable of addressing a broad spectrum of application requirements—from low-cost hobbyist projects to high-reliability industrial systems.

The STM32 family is segmented into multiple sub-families, each targeting specific needs related to performance, power consumption, integration level, and peripheral availability. The most significant and commonly used STM32 sub-families include STM32F0, F1, F3, F4, F7, L0, L4, H7, G0, G4, WB, and WL series. These microcontrollers cover all performance levels, ranging from low-power 8-bit MCU alternatives to high-performance DSP-capable 32-bit MCUs, all while sharing a consistent software and development ecosystem.

STM32F1 Series (Mainstream MCUs):
The STM32F1 series was the first in the STM32 family and remains one of the most widely used families due to its reliability and performance. Based on the ARM Cortex-M3 core, it offers a balanced feature set and is ideal for general-purpose applications. The F1 series introduced developers to a 32-bit alternative to traditional 8- and 16-bit MCUs, offering significantly more processing power, larger memory capacities, and better peripheral integration.

The STM32F103 devices, including the popular STM32F103C8T6 variant used in the Blue Pill board, are notable for their clock speeds of up to 72 MHz, wide range of timers, multiple USARTs, SPI and I2C buses, and ADCs. These features, combined with its low cost, have made the STM32F1 series a favorite in academic labs, maker communities, and low-cost industrial products.

STM32F0 and F3 Series (Cost-Effective and Analog-Enhanced MCUs):
To address the demand for cost-sensitive applications, ST introduced the STM32F0 series, which utilizes the ARM Cortex-M0 core. It is designed to provide a smooth migration path for developers accustomed to 8-bit architectures while benefiting from 32-bit performance and features. The F0 series supports essential peripherals and basic timers, making it ideal for simple control and sensor interface applications.
The STM32F3 series, on the other hand, is based on the Cortex-M4 core and is optimized for mixed-signal applications.

STM32F4 and F7 Series (High-Performance DSP-Capable MCUs): As embedded applications demanded more processing power, the STM32F4 series was introduced, featuring the ARM Cortex-M4 core running at up to 180 MHz and supporting floating-point operations. It integrates rich peripherals, larger flash and RAM sizes, and enhanced DMA and memory controllers. These microcontrollers are used in applications involving real-time audio processing, motor control, and data acquisition systems that require significant computational capabilities.

Building upon the F4, the STM32F7 series brought even higher performance with the ARM Cortex-M7 core, running at up to 216 MHz. The F7 family is intended for applications requiring high-speed processing and graphical user interfaces, such as digital oscilloscopes, wearable electronics, and multimedia devices.

STM32L Series (Ultra-Low Power MCUs): For applications where power efficiency is paramount, such as in portable or battery-operated devices, STMicroelectronics developed the STM32L series. This family includes STM32L0 (Cortex-M0+), L1 (Cortex-M3), L4 (Cortex-M4), and L5 (Cortex-M33) subseries. These MCUs offer various low-power modes, dynamic voltage scaling, and optimized wake-up times to enable long-term autonomous operation with minimal energy usage.
The L series supports a wide range of features including capacitive touch sensing, USB, LCD controllers, and integrated security modules. These MCUs are prevalent in medical devices, remote sensing, and wearable technology.

STM32G0 and G4 Series (Next-Generation General-Purpose MCUs):
The STM32G0 and G4 series represent a new generation of general-purpose and real-time control MCUs. The G0 series is designed to replace legacy 8-bit MCUs, offering better performance, security, and integration with a Cortex-M0+ core. It focuses on affordability without compromising modern features like hardware security, power management, and analog integration.
The STM32G4, equipped with a Cortex-M4 core, is aimed at high-performance real-time applications such as digital power supplies and motor control. It features advanced timers, DACs, comparators, and high-resolution ADCs, making it well-suited for demanding analog-centric applications.

STM32H7 Series (High-Performance Dual-Core MCUs): For applications requiring maximum computational power and memory bandwidth, the STM32H7 series is ST's flagship MCU line. These devices feature the ARM Cortex-M7 core running at up to 480 MHz, often coupled with a secondary Cortex-M4 core for parallel processing. With features like large on-chip RAM, advanced graphics processing, hardware cryptographic accelerators, and rich communication peripherals, the H7 series bridges the gap between traditional microcontrollers and microprocessors (MCUs vs. MPUs).

The H7 series is often deployed in applications such as industrial control, AI/ML at the edge, and advanced medical instrumentation where real-time performance and security are critical.

2.2 Key Features of the STM32F103C8T6 (Blue Pill):

The STM32F103C8T6, widely recognized as the "Blue Pill," stands out in the world of embedded systems due to its excellent balance of performance, peripheral richness, and affordability. Powered by the 32-bit ARM Cortex-M3 core running at a frequency of up to 72 MHz, it delivers efficient computational capability while maintaining low power consumption—making it suitable for both real-time and low-energy applications. The microcontroller includes 64 KB of Flash memory and 20 KB of SRAM, providing ample space for moderately complex firmware and runtime data operations. A key strength of the STM32F103C8T6 lies in its extensive peripheral set, which includes up to 37 general-purpose I/O pins, multiple timers, and two 12-bit ADCs capable of sampling analog signals at high resolution—essential for applications involving environmental sensing, condition monitoring, or analog signal processing.

Communication flexibility is another standout feature, with built-in support for USART, SPI, I2C, CAN, and USB interfaces. These enable reliable and high-speed communication with external modules such as sensors, actuators, memory devices, or wireless transceivers like LoRa and Bluetooth. The board also supports advanced real-time features such as Direct Memory Access (DMA), enabling peripherals to transfer data to and from memory without CPU intervention, significantly enhancing performance in data-intensive applications. Additionally, the built-in clock management system, including PLL-based clock multiplication, allows for flexible timing configurations. Power management capabilities such as sleep, stop, and standby modes ensure efficient energy use, which is critical in battery-operated or remote systems.

The STM32F103C8T6, widely recognized as the "Blue Pill," stands out in the world of embedded systems due to its excellent balance of performance, peripheral richness, and affordability.

2.3 Role of ARM Cortex-M3 in Embedded Application

The ARM Cortex-M3 processor plays a pivotal role in modern embedded system design due to its efficient architecture, real-time processing capabilities, and broad ecosystem support. Developed by ARM Holdings, the Cortex-M3 is specifically optimized for low-power, high-performance applications that require deterministic behavior—making it exceptionally well-suited for use in microcontrollers like the STM32F103C8T6. Its popularity in embedded applications is largely due to the balance it offers between

processing power, code density, interrupt latency, and power consumption, all of which are critical factors in designing responsive and reliable embedded systems.

of the defining characteristics of the Cortex-M3 is its 32-bit Harvard architecture with a three-stage pipeline, allowing it to execute instructions efficiently while minimizing power usage. With support for the Thumb-2 instruction set, the Cortex-M3 combines 16-bit and 32-bit instructions to optimize memory usage and improve code density—an important feature in memory-constrained environments. This makes it ideal for firmware development in microcontrollers with limited flash and SRAM capacities, such as the 64KB Flash and 20KB RAM configuration of the STM32F103C8T6. The processor also includes a hardware divide instruction and single-cycle multiplication, which greatly improves the performance of mathematical operations essential for digital signal processing, control systems, and real-time analytics.

A cornerstone feature of the Cortex-M3 is its highly efficient Nested Vectored Interrupt Controller (NVIC). The NVIC supports low-latency interrupt handling with up to 256 priority levels and dynamic interrupt nesting. This capability is crucial in embedded systems where multiple sensors, communication interfaces, and real-time tasks must operate concurrently without interfering with each other. The deterministic interrupt response time of the Cortex-M3 ensures that high-priority tasks such as motor control loops, signal acquisition, and emergency shutdown procedures are handled promptly and predictably.

From a hardware integration perspective, the Cortex-M3 is designed to seamlessly interface with a wide range of peripherals and memory structures through its AMBA AHB-Lite bus architecture. This ensures high-speed communication between the core and memory/peripherals, enabling efficient data transfer and system responsiveness. The inclusion of bit-banding allows atomic manipulation of individual bits in memory, which is especially valuable in safety-critical and low-level hardware control applications such as setting digital outputs or managing flags.

The Cortex-M3’s interrupt handling capabilities are also highly refined, largely due to its built-in Nested Vectored Interrupt Controller (NVIC). The NVIC supports dynamic prioritization and nesting of up to 256 interrupt levels, allowing for complex multitasking and real-time responsiveness without the overhead traditionally associated with interrupt processing. This is particularly important in embedded systems where time-sensitive operations such as motor control, communication protocol handling, or sensor sampling must be prioritized and managed efficiently. Low interrupt latency ensures that critical operations can pre-empt less urgent ones, leading to more reliable and deterministic behaviour, which is essential in mission-critical systems such as medical monitoring or aerospace electronics.

Another notable feature of the Cortex-M3 is the integrated System Tick Timer (SysTick), a 24-bit timer that can be used to create periodic interrupts. This feature is commonly used to implement task schedulers or support real-time operating systems (RTOS) like FreeRTOS or RTX, enabling embedded systems to manage multiple threads or processes in a predictable and coordinated manner. Many commercial and open-source RTOS platforms have native support for the Cortex-M3 architecture, providing developers with access to real-time task management, inter-process communication, and time-triggered operations all of which are crucial in building scalable embedded systems. For instance, a remote environmental monitoring system might require simultaneous temperature sensing, data logging, and wireless transmission, all of which can be effectively handled through RTOS multitasking on a Cortex-M3-based MCU.

The ARM Cortex-M3 processor has emerged as a cornerstone in the development of modern embedded systems, especially in applications that demand real-time responsiveness, low power consumption, and cost-effectiveness. Introduced by ARM Holdings as part of the ARMv7-M architecture, the Cortex-M3 was designed with embedded and low-power applications in mind, offering features that cater to industrial automation, medical devices, consumer electronics, Internet of Things (IoT) applications, and many other domains. At its core, the Cortex-M3 combines a 32-bit RISC architecture with a simplified instruction set and hardware optimizations that allow developers to build compact yet high-performance embedded solutions. Its wide adoption by leading microcontroller manufacturers, such as STMicroelectronics in their STM32F1 series, including the STM32F103C8T6 (Blue Pill), is a testament to the processor's versatility and capability.

One of the key strengths of the Cortex-M3 is its efficient instruction set architecture (ISA), known as Thumb-2. This hybrid instruction set combines the high performance of 32-bit instructions with the reduced memory footprint of 16-bit instructions, optimizing both speed and code density. As embedded systems often operate under tight memory constraints, this feature is particularly

Power efficiency is another defining characteristic of the ARM Cortex-M3 processor. It is designed to operate in multiple low-power modes, including sleep, stop, and standby, which allow systems to significantly reduce energy consumption during idle periods. This makes the Cortex-M3 ideal for battery-powered and remote-sensing applications, where power conservation is paramount. For example, in an IoT device using a Cortex-M3 MCU, the system can wake up periodically to read sensors and transmit data, then return to a low-power state, extending battery life without sacrificing functionality. This ability to manage power intelligently aligns well with the global shift toward sustainable and energy-efficient electronic systems.

From a software development perspective, the Cortex-M3 is exceptionally well supported. Developers can choose from a wide array of integrated development environments (IDEs) such as Keil MDK, IAR Embedded Workbench, and STMicroelectronics' own STM32CubeIDE. These tools provide robust support for code generation, debugging, and performance profiling. Open-source platforms like PlatformIO and STM32duino also offer flexible alternatives for hobbyists and professionals alike. The presence of comprehensive development libraries, such as CMSIS (Cortex Microcontroller Software Interface Standard) and the STM32 HAL (Hardware Abstraction Layer), allows developers to work at various levels of abstraction—from low-level register control to high-level peripheral configuration. This ecosystem accelerates development cycles and reduces the time to market for embedded solutions.

One of the primary features that make the Cortex-M3 ideal for embedded applications is its Thumb-2 instruction set architecture, which provides a mix of 16-bit and 32-bit instructions. This hybrid instruction set allows developers to achieve high code density without sacrificing performance. In constrained embedded systems—such as those with limited flash and RAM—this ability to reduce the memory footprint is crucial. The STM32F103C8T6, for example, offers 64 KB of flash memory and 20 KB of SRAM, and the Cortex-M3's optimized instruction set allows efficient use of this memory. Furthermore, the processor includes a three-stage pipeline and hardware multiply/divide units, enabling it to execute many instructions in a single clock cycle. This results in fast processing of control algorithms, signal processing, and data handling tasks, which are core requirements in embedded applications ranging from motor control to real-time sensor data acquisition.

Another significant advantage of the Cortex-M3 is its sophisticated interrupt handling system. At the heart of this capability is the Nested Vectored Interrupt Controller (NVIC), which allows for low-latency, prioritized interrupt servicing. The NVIC supports up to 240 external interrupts with programmable priority levels, enabling efficient context switching and rapid responses to external events. This feature is critical in real-time application

2.4 Development Environments and Community Support

In the rapidly evolving world of embedded systems, the availability and quality of development environments, tools, and community support are just as critical as the hardware itself. The STM32F103C8T6 microcontroller, known in the embedded community as the "Blue Pill," has gained significant traction not only due to its technical specifications and cost-efficiency but also because of the extensive ecosystem that supports it. This ecosystem includes integrated development environments (IDEs), software libraries, programming tools, documentation, and a vibrant global developer

community. Together, these resources enable developers—from beginners to professionals—to rapidly prototype, test, and deploy embedded applications across a wide range of industries.

One of the cornerstone tools provided by STMicroelectronics for STM32 development is the STM32CubeIDE, a comprehensive IDE based on Eclipse and GNU Arm toolchain integration. This IDE combines code editing, debugging, peripheral configuration, and compilation into a single platform, offering an efficient workflow for embedded software engineers. A powerful feature within this ecosystem is STM32CubeMX, a graphical software configuration tool that facilitates pin assignment, clock setup, and middleware integration for the target STM32 device. Developers can visually select and configure peripherals, generate initialization code, and create project files compatible with various IDEs. This reduces the learning curve significantly and minimizes manual coding errors during peripheral setup, thus accelerating project development.

Complementing STM32CubeIDE, STMicroelectronics also provides firmware packages, including HAL (Hardware Abstraction Layer) and LL (Low-Level) drivers. The HAL offers high-level, user-friendly APIs that simplify the process of writing embedded software, while the LL drivers provide more direct access to the hardware registers for performance-optimized applications. Developers can choose between these layers depending on their familiarity, required performance, and desired control over hardware. The modularity of these libraries ensures that the same code can often be reused across different STM32 devices with minor modifications, enhancing scalability and maintainability of applications.

Beyond ST's official tools, a wide range of third-party development environments is compatible with the STM32F103C8T6. The Keil MDK-ARM (by Arm) and IAR Embedded Workbench (by IAR Systems) are popular commercial-grade IDEs offering professional-grade debugging, simulation, and code optimization features. These environments are widely used in industry for developing safety-critical and high-performance embedded application

On the open-source front, the GNU Arm Embedded Toolchain provides a cost-free and highly flexible toolchain for compiling, linking, and debugging STM32 firmware. It integrates well with Visual Studio Code, a modern, extensible code editor. Using extensions like Cortex-Debug and PlatformIO, developers can enjoy features such as IntelliSense, source-level debugging, and real-time watch windows, rivaling the functionality of commercial IDEs. PlatformIO, in particular, provides an advanced cross-platform development environment that supports multiple boards, frameworks (Arduino, STM32Cube, mbed), and build systems, making it an ideal solution for collaborative and automated development environments.

A key enabler of STM32 development is the support for multiple programming and debugging interfaces. The Blue Pill board can be programmed via ST-Link V2, USB-to-Serial adapters (such as FTDI), or through DFU (Device Firmware Upgrade) mode via USB. This flexibility is particularly beneficial for users with limited access to professional tools. Moreover, flashing utilities like STM32CubeProgrammer, OpenOCD, and Serial Loader give developers multiple options for uploading firmware, verifying memory contents, and even performing in-system programming without removing the microcontroller from the target board.

For beginners and rapid prototyping, the Arduino IDE presents a simplified entry point. Through the STM32duino project, STM32 microcontrollers, including the Blue Pill, are made compatible with the Arduino programming model. This allows developers to write code using simplified Arduino functions, while still accessing advanced STM32 features under the hood. Libraries for I2C, SPI, UART, ADC, and digital IO operations are readily available, allowing for quick implementation of common tasks such as reading sensors, driving displays, or controlling motors. Additionally, the Arduino ecosystem provides a vast repository of user-contributed libraries and example sketches, which further accelerate learning and experimentation.

The presence of an active and enthusiastic global community has greatly contributed to the widespread adoption of the STM32F103C8T6. Online platforms such as the ST Community Forum, Stack Overflow, GitHub, Reddit's r/embedded, and specialized blogs like How2Electronics and Embedded Lab offer a continuous stream of shared knowledge, code examples, design insights, and troubleshooting assistance. This global collaborative environment allows developers to resolve challenges more efficiently, learn best practices, and keep up with emerging trends. Countless GitHub repositories host open-source projects using the Blue Pill for robotics, home automation, IoT, wearable tech, and industrial monitoring—offering inspiration and reference designs for new developers.

CHAPTER 3: SYSTEM IMPLEMENTATION

The system implementation phase is one of the most crucial stages in the lifecycle of any embedded systems project. It represents the transformation of theoretical concepts, schematic designs, and planning documents into a tangible, functional solution. At this stage, all elements—hardware components, microcontroller firmware, interfaced peripherals, and communication protocols—are cohesively brought together to construct a working prototype or final product. For this particular project, the implementation is centered around the STM32F103C8T6 microcontroller, commonly referred to as the "Blue Pill," which serves as the central processing unit of the entire system.

In embedded system development, implementation goes beyond merely assembling hardware. It involves carefully integrating both the physical and logical aspects of the system to ensure they work in harmony. This includes configuring input/output pins, initializing communication interfaces, developing real-time firmware, and ensuring all subsystems operate within the defined timing and functional constraints. Unlike general-purpose computing, embedded systems often operate in time-sensitive environments where performance and reliability are non-negotiable. Therefore, the implementation must be conducted with precision, resource awareness, and foresight for scalability.

This chapter elaborates on the methodologies, design strategies, tools, and engineering practices used to successfully implement the embedded application. The chapter begins with a detailed breakdown of the hardware setup, including the interfacing of sensors, power supply considerations, signal conditioning, and microcontroller pin configurations. The goal of the hardware implementation is to establish a robust and stable foundation capable of supporting the intended features of the system without electrical or mechanical failure. Various design factors such as voltage levels, current requirements, mechanical placement, and environmental conditions are taken into account during this stage.

Following the hardware section, the focus shifts to firmware development, which encompasses writing, debugging, and optimizing the software that controls the STM32 microcontroller. This process includes peripheral initialization, interrupt configuration, data acquisition routines, control algorithms, and communication handlers. Firmware is written in C/C++ using industry-standard IDEs such as STM32CubeIDE or Arduino-based environments, depending on the complexity and project goals. Special attention is given to structuring the code efficiently to ensure modularity, readability, and maintainability. Tools such as STM32CubeMX aid in visual peripheral configuration and code generation, streamlining the setup of essential modules like ADC, USART, and Timers.

Additionally, this chapter discusses how the system was tested and validated throughout the implementation phase. Debugging tools such as ST-Link V2, serial terminals, and logic analyzers were utilized to identify and correct both hardware and software issues. Integration testing ensured that each subsystem worked as expected when combined. Timing analysis and power consumption profiling were performed where applicable to ensure that the system meets both functional and non-functional requirements, such as low power operation or real-time responsiveness.

3.1 System Overview

The system overview provides a comprehensive summary of the embedded solution's architecture, components, and operation. It serves as a high-level representation of how different hardware and software modules interact to achieve the intended functionality. In this project, the embedded system is built around the STM32F103C8T6 microcontroller, which acts as the central processing unit responsible for collecting, processing, and transmitting data from various peripherals. The design goal is to develop a reliable, scalable, and real-time embedded platform that can operate autonomously while communicating critical data wirelessly or displaying results locally.

The system comprises several essential modules: a power supply unit, a sensing unit (which may include temperature, humidity, vibration, or motion sensors), a processing unit (the STM32 microcontroller itself), a communication module (such as LoRa, Wi-Fi, or Bluetooth), and an optional display or user interface. Each of these blocks plays a vital role in the overall operation of the system and is carefully designed and integrated to ensure performance under real-time constraints.

At the heart of the system is the STM32F103C8T6, a 32-bit ARM Cortex-M3-based microcontroller known for its balance of computational power and energy efficiency. With a clock speed of up to 72 MHz, built-in Flash and SRAM memory, and numerous I/O and communication peripherals, it is capable of handling multiple tasks such as sensor data acquisition, interrupt-driven communication, and real-time control simultaneously. Its on-chip peripherals such as ADCs, timers, and UART/SPI/I2C interfaces allow seamless interaction with both analog and digital devices.

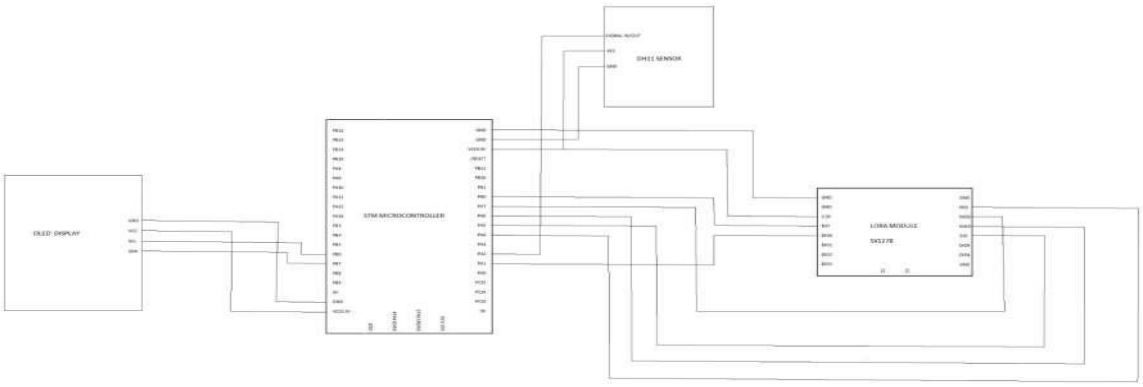
The sensing unit is responsible for capturing data from the environment. Depending on the specific application, sensors such as DHT11 (for temperature and humidity), MQ gas sensors, accelerometers, or Hall effect sensors are connected to the microcontroller via analog or digital input pins.

3.2 Hardware Setup and Pin Configuration

The hardware setup of the proposed system forms the foundational layer upon which all sensing, communication, and control tasks are performed. This section outlines the detailed physical configuration of each component used in the project, describes the purpose of each hardware element, and explains how the STM32F103C8T6 (Blue Pill) microcontroller interfaces with these components to enable a functioning low-power, long-range LoRa communication system.

System Description: The project is designed to transmit environmental sensor data (e.g., from a DHT11 sensor) over long distances using LoRa SX1276 (915 MHz) transceivers, controlled by STM32F103C8T6 microcontrollers on both the sender and receiver sides. Supporting components like an OLED display, power circuitry, and connecting wires are integrated on a solderless breadboard for prototyping.

Circuit Diagram:



The circuit schematic reflects the physical interconnections between the microcontroller and the peripheral devices. Each component is connected using standard protocols (SPI, I2C, digital I/O) and powered via onboard regulated voltage sources. LoRa SX1276 Module (915 MHz):

- Connections to STM32 (SPI):
  - NSS (CS) → PA4 (GPIO Output)
  - SCK → PA5 (SPI Clock)

    - MISO → PA6 (SPI Master In Slave Out)
    - MOSI → PA7 (SPI Master Out Slave In
    - RST → PB0 (GPIO Output for module reset)
    - DIO0 → PA1 (Digital input used for interrupts/data ready)
- Power: 3.3V from the Blue Pill's onboard regulator; ensure common ground (GND).

The LoRa module uses SPI for fast and reliable data exchange. The NSS pin is used to initiate and terminate communication sessions. The RST pin is toggled at startup or reset via firmware. DIO0 is used as an interrupt line to notify the STM32 when a message has been received.

Power Supply & Reset Circuit:

- Powering the STM32:
    - The board can be powered via the USB port, Vin pin (5V), or a battery supply.
    - The onboard AMS1117 regulator converts 5V to 3.3V for MCU and peripheral operation.
- Reset Button:
    - Connected to the NRST pin with a 10kΩ pull-up resistor to 3.3V.
    - Allows manual hardware reset during development or recovery from faults.
- BOOT0 Pin:
    - Pulled LOW with a resistor to ensure the microcontroller boots from flash memory.

Peripheral	STM32 Pin	Direction	Purpose
LoRa NSS (CS)	PA4	Output	SPI Chip Select
LoRa RST	PB0	Output	LoRa Reset Pin
LoRa DIO0 (IRQ)	PA1	Input	Interrupt from LoRa
SPI SCK	PA5	Output	Clock for SPI
SPI MISO	PA6	Input	SPI Master In
SPI MOSI	PA7	Output	SPI Master Out
DHT11 Data	Any GPIO (e.g., PB1)	Bidirectional	Digital sensor data line
OLED SDA	PB7	Bidirectional	I2C Data
OLED SCL	PB6	Output	I2C Clock
UART TX (debug)	PA9	Output	Serial communication to PC

UART RX (debug)	PA10	Input	Serial receive from PC

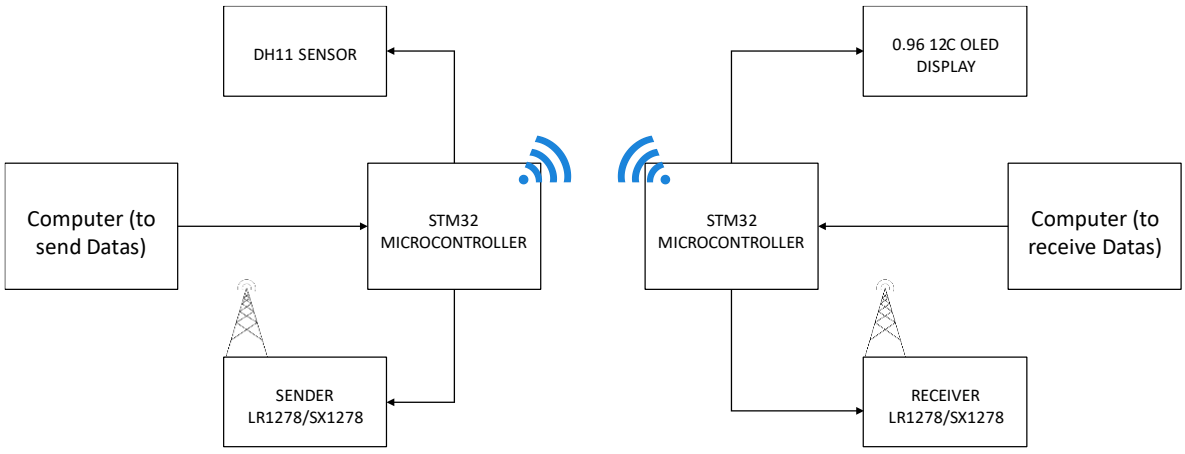
The system is divided into two logical blocks: the Transmitter Unit and the Receiver Unit. Both units are centered around the STM32F103C8T6 microcontroller, which is responsible for handling all input/output operations, processing, and communication.
Transmitter Unit Block Diagram:

- DHT11 Sensor → STM32F103C8T6 → LoRa SX1276 Module
  In this section of the system, the DHT11 sensor collects temperature and humidity data from the environment. This data is then read by the STM32 using a digital GPIO pin. The microcontroller formats the data into a structured packet and sends it to the LoRa SX1276 module using the SPI protocol. The LoRa module then transmits the data wirelessly to the receiver unit over the 915 MHz frequency band.

Receiver Unit Block Diagram:

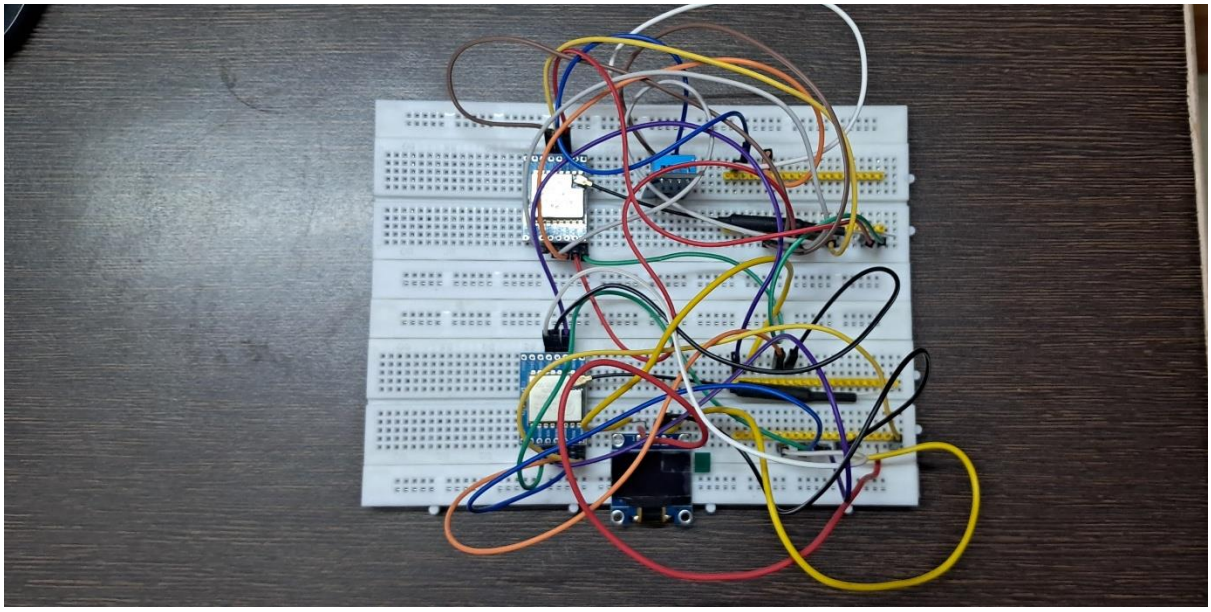
- LoRa SX1276 Module → STM32F103C8T6 → OLED Display / Serial Monitor
  The LoRa module at the receiver end receives the transmitted data and passes it to the STM32 via SPI. The microcontroller then extracts the sensor values from the data packet and displays it on a 0.96" OLED screen using I2C communication. Optionally, the same data can be sent to a PC over a USB serial connection for logging or visualization on a terminal application such as Tera Term or PuTTY.

This modular architecture allows easy expansion of the system—for example, more sensors can be added to the transmitter or a data logger can be integrated into the receiver.



3.3 Prototype Hardware Setup:

The image above shows the assembled prototype of the receiver unit for the LoRa-based embedded communication system. It features an STM32F103C8T6 (Blue Pill) microcontroller mounted on a solderless breadboard, connected to a LoRa SX1276 module via SPI for wireless data reception. An OLED display is also connected using the I2C interface to visualize incoming sensor data. The wiring includes color-coded jumper wires for clarity and ease of debugging. Power is supplied through a regulated 3.3V source via USB or external power rail. This physical layout allows modular testing and validation of communication, display, and sensor integration features in a real-world environment.



Peripheral Interfaces: GPIO, ADC, UART, SPI, I2C

In embedded systems, peripheral interfaces act as the vital channels through which the microcontroller communicates with external components such as sensors, displays, actuators, and communication modules. The STM32F103C8T6 microcontroller, based on the ARM Cortex-M3 architecture, offers a rich set of hardware interfaces that provide flexibility and efficiency in system design. These interfaces include General Purpose Input/Output (GPIO), Analog-to-Digital Converter (ADC), Universal Asynchronous Receiver-Transmitter (UART), Serial Peripheral Interface (SPI), and Inter-Integrated Circuit (I2C). Each of these peripherals plays a distinct and crucial role in ensuring effective communication and control between the microcontroller and its surrounding hardware environment.

General Purpose Input/Output (GPIO): The GPIO pins of the STM32F103C8T6 are among the most fundamental interfaces used in nearly every embedded application. These pins are versatile and can be configured as either digital inputs or outputs. In this project, GPIO pins serve several key purposes. For example, certain GPIOs are configured as digital outputs to control modules like LoRa (e.g., the NSS/CS and RESET pins), while others function as digital inputs to receive signals from modules like the DHT11 sensor or to handle interrupt lines such as the DIO0 pin on the LoRa module. GPIOs are also used for reading push-buttons or toggling indicator LEDs for debugging and status signaling. The STM32 provides programmable pull-up and pull-down resistors, enabling noise-immune input configurations, and supports external interrupt triggering on edge or level changes, which is particularly useful in time-sensitive applications. The ability to configure the mode, speed, and pull settings of each GPIO pin gives developers precise control over how signals are managed in both high-speed and low-power applications.

Analog-to-Digital Converter (ADC): The Analog-to-Digital Converter is another essential peripheral used in sensor-based systems where real-world analog signals must be digitized for processing. The STM32F103C8T6 includes two 12-bit ADCs capable of sampling up to 1 million samples per second (1 Msps), supporting single, continuous, scan, and injected conversion modes. Although the current implementation focuses on digital sensors like the DHT11, which output already-digitized data, the ADC becomes highly relevant for interfacing with analog sensors such as temperature probes (LM35), potentiometers, or voltage monitoring circuits.

Universal Asynchronous Receiver/Transmitter (UART): UART is a widely used asynchronous serial communication protocol, and it is integral to this project for debugging and monitoring. The STM32F103C8T6 provides multiple USART/UART peripherals, with PA9 (TX) and PA10 (RX) commonly used for standard serial communication. During development and testing, the UART interface connects the microcontroller to a computer via a USB-to-Serial converter, enabling bidirectional communication with terminal software like Tera Term, PuTTY, or Arduino Serial Monitor.
This serial link is used to output debug logs, display received sensor data, and monitor communication between nodes. In future applications, this UART channel can also be redirected for communication with GSM modules, GPS receivers, or Bluetooth modules. STM32's UART peripherals support various features including hardware flow control, interrupt-driven data handling, and DMA (Direct Memory Access) for efficient data transfer, all of which enhance performance in high-throughput communication scenarios.

The STM32F103C8T6 microcontroller supports a variety of essential peripheral interfaces that enable seamless integration with sensors, communication modules, and display devices. GPIOs are used for basic digital input/output functions, such as reading sensor states or controlling module resets. The ADC provides a way to interface with analog sensors, offering precise 12-bit digital conversion of real-world signals. UART facilitates serial communication, commonly used here for debugging or data transmission to a PC terminal.

3.4 Programming and Debugging Tools (STM32CubeIDE, ST-Link, USB-TTL)

Programming and debugging are critical aspects of embedded system development, as they directly influence the reliability, performance, and maintainability of the system. Efficient tools streamline the development process, minimize errors, and provide insights into real-time execution. For this project, which utilizes the STM32F103C8T6 (commonly known as the "Blue Pill"), three main tools were employed for programming and debugging: STM32CubeIDE, ST-Link V2 programmer, and a USB-to-TTL (UART) converter. These tools form a complete development ecosystem that enables writing, uploading, testing, and troubleshooting embedded firmware.

STM32CubeIDE: STM32CubeIDE, developed by STMicroelectronics, is an integrated development environment (IDE) based on Eclipse and the GNU Arm toolchain. It provides a unified platform for writing code, configuring peripherals, compiling, flashing, and debugging firmware for STM32 microcontrollers. One of the key benefits of STM32CubeIDE is its seamless integration with STM32CubeMX, a graphical configuration tool that simplifies microcontroller setup. Using CubeMX within the IDE, developers can visually assign pin functions, enable peripherals (such as SPI, I2C, UART, ADC), configure system clocks, and generate initialization code all of which are crucial for setting up a reliable embedded project.

In this project, STM32CubeIDE was used to develop the core firmware for both the transmitter and receiver nodes. The firmware includes sensor data acquisition (e.g., reading DHT11), wireless communication using LoRa (via SPI), and display handling through I2C. The HAL (Hardware Abstraction Layer) libraries provided by ST were used to interact with low-level hardware components, allowing modular and maintainable code development. STM32CubeIDE also supports powerful debugging features such as variable monitoring, breakpoints, memory inspection, and single-step execution—all accessible via its graphical interface.

Additionally, STM32CubeIDE supports project version control and multi-core builds, enhancing collaboration in larger development teams. Its console output and debugging

terminal allow developers to trace program execution and identify logic or peripheral configuration errors efficiently. The auto-generated code structure ensures organized and scalable code development, which is particularly helpful in embedded systems that evolve to include new modules or sensors.

ST-Link V2 Debugger and Programmer: To program the STM32F103C8T6 microcontroller with the firmware developed in STM32CubeIDE, a ST-Link V2 programmer/debugger was used. The ST-Link is an official programming tool provided by STMicroelectronics that supports both SWD (Serial Wire Debug) and JTAG interfaces. For STM32F1 series devices like the Blue Pill, the SWD interface is preferred due to its simplicity and minimal pin usage (requiring only SWDIO, SWCLK, and GND).
In this project, the ST-Link V2 was connected to the Blue Pill using four pins:

- SWDIO (Data I/O)
- SWCLK (Clock)
- 3.3V (Power Reference)
- GND (Ground)

Once connected, the ST-Link allows users to flash firmware, perform in-circuit debugging, and even recover microcontrollers that have been soft-bricked due to faulty firmware. It integrates directly with STM32CubeIDE, which automatically detects the device, uploads the compiled binary, and initiates the debugging session. This tight integration makes it extremely efficient to detect run-time faults, check peripheral registers, and step through code line-by-line.

Furthermore, ST-Link provides access to advanced debugging features like:

- Setting hardware breakpoints at specific instruction addresses.
- Inspecting live values of global and local variables.
- Viewing processor core registers and system memory.
- Analyzing exception handling and system crashes.

These features are invaluable in complex systems where timing, logic flow, or hardware synchronization issues are difficult to diagnose without in-circuit observation. The ST-Link's reliability, speed, and support for a wide range of STM32 devices make it a go-to tool for professional-grade development and testing.

USB-to-TTL (UART) Converter: While ST-Link is essential for flashing and debugging, the USB-to-TTL converter plays a complementary role, especially in serial communication and debugging output. The STM32F103C8T6 includes multiple USART peripherals that allow data to be transmitted to and received from external devices or terminals via the UART protocol. In this project, PA9 (TX) and PA10 (RX) are used to communicate with the USB-to-TTL module.

The USB-to-TTL converter enables the developer to connect the STM32 to a computer via a USB port and monitor UART output using terminal software like Tera Term, PuTTY, or Arduino Serial Monitor. This allows real-time display of sensor readings, LoRa messages, debug logs, or error reports. It is especially useful when testing code changes or verifying that data is correctly received from the LoRa module and displayed by the receiver.

Moreover, USB-TTL converters can be used to upload firmware using STM32's built-in bootloader mode (via the BOOT0 pin), offering an alternative programming path when ST-Link is not available. This method requires setting BOOT0 to HIGH during reset, which instructs the MCU to enter system memory bootloader and receive code over UART. Though not used as the primary method in this project, this functionality adds resilience and flexibility to the development process.

The combination of STM32CubeIDE, ST-Link V2, and USB-to-TTL modules creates a robust and versatile programming and debugging environment. STM32CubeIDE serves as the primary firmware development and peripheral configuration platform, offering deep integration with ST hardware and toolchains. The ST-Link programmer ensures seamless firmware uploading and professional-level debugging, while the USB-to-TTL converter adds a real-time monitoring channel that aids during runtime verification and troubleshooting.

3.5 Sample Applications: LED Blink, Sensor Interfacing, Serial Communication:

To verify the functionality of the STM32F103C8T6 microcontroller and its peripheral configurations, several basic sample applications were implemented and tested. These sample programs serve as foundational exercises in embedded system development and are used to validate hardware setups, check pin configurations, and ensure correct operation of software libraries and toolchains. Among the most essential and illustrative applications developed are the LED blink, sensor interfacing, and serial communication programs. These tests confirm the working condition of GPIOs, timers, ADCs or digital input protocols, and USART peripherals. Although simple in appearance, they provide invaluable feedback during the early phases of system prototyping.

The LED blink application is often regarded as the “Hello World” of embedded systems. It is typically the first program uploaded to a microcontroller to verify that it is correctly powered, programmable, and functioning. In this project, the onboard or externally connected LED was interfaced with a GPIO pin configured as digital output—for example, PC13, which is commonly connected to the onboard LED in STM32 Blue Pill boards.

In the firmware, the GPIO pin is first initialized using the STM32CubeMX graphical configurator or manually in code. The output mode is set to push-pull, and the speed is configured to low or medium. A simple infinite loop toggles the GPIO pin's state with a delay introduced between transitions, resulting in a visible blinking effect. The delay is implemented using software loops or, more accurately, hardware timers for improved timing precision. This test confirms the proper configuration of system clocks, pin mapping, and the correctness of the compilation and upload process via ST-Link.

Apart from verifying basic functionality, the LED blink test is used later for status indication, where different blink rates or patterns can be used to represent various system states such as successful initialization, error detection, or message reception. Thus, even a basic program like LED blink evolves into a valuable debugging and feedback mechanism in more advanced stages of the project.

The next step in validating the STM32's functionality is interfacing it with real-world sensors. In this project, a DHT11 digital temperature and humidity sensor was used on the transmitter side to collect environmental data. The DHT11 communicates over a single-wire digital protocol and outputs data packets in a time-sensitive format. To interface this sensor with the STM32F103C8T6, a GPIO pin (e.g., PB1) is configured as an open-drain digital I/O and is used both to send the start signal to the sensor and to receive the response data.

The firmware involves initiating a start condition by pulling the data line low for a specific duration, then switching the GPIO to input mode to capture the bitwise response sent by the sensor. Due to the strict timing requirements of the DHT11 protocol, the STM32's microsecond-level delay capabilities are leveraged using SysTick or hardware timers. The received 40-bit data stream is parsed into temperature, humidity, and checksum values. These values are stored in variables and optionally displayed over UART or OLED.

The successful integration of the DHT11 confirms the STM32's ability to manage real-time digital protocols and handle GPIO direction switching within tight timing constraints. This sensor interfacing application serves as the foundation for integrating more complex or multiple sensors in future iterations, such as gas sensors, light sensors, or analog-based modules using the onboard ADC

Serial Communication: UART Debugging and Data Logging: Serial communication plays a critical role in embedded systems for data exchange, debugging, and configuration. In this project, the Universal Asynchronous Receiver/Transmitter (UART) interface was used to send and receive data between the STM32F103C8T6 and a personal computer. This connection was established using a USB-to-TTL converter

(such as CP2102 or FTDI), interfaced through the STM32's PA9 (TX) and PA10 (RX) pins.

The application involves initializing the USART peripheral at a standard baud rate, such as 9600 or 115200 bps, and sending formatted strings over the serial port. The transmitted data includes sensor readings from the DHT11 module, system status messages, and diagnostic feedback. On the PC side, terminal software like Tera Term, PuTTY, or Arduino Serial Monitor is used to monitor the data stream. This enables developers to validate whether the correct data is being read, formatted, and transmitted, and to track down issues in real-time during code execution.

UART communication also enables the implementation of command-response protocols, where the system receives input commands from the user and returns corresponding actions or data. For example, a character sent from the serial terminal may trigger the STM32 to re-read the sensor, toggle an LED, or return its firmware version. This interactive capability is particularly valuable for field diagnostics or future enhancements where dynamic user interaction is required.

Building upon these basic implementations, each sample application serves as a stepping stone toward integrating more advanced features within the system. For example, the LED blink test can be enhanced using PWM (Pulse Width Modulation) to create variable brightness control, which may be useful for indicating sensor thresholds or battery levels in IoT devices. Similarly, the DHT11 sensor interface can be expanded to support multiple sensors or replaced with higher-precision modules such as DHT22 or BME280, demonstrating the STM32's scalability and flexibility in accommodating various sensor protocols.

3.6 Data Transmission and Communication Protocols:

Efficient and reliable data transmission is one of the most critical aspects of embedded systems, particularly in applications involving remote sensing, monitoring, and control. The success of such systems depends heavily on the selection and implementation of appropriate communication protocols that ensure timely, error-free, and energy-efficient exchange of information. In this project, data transmission is achieved through the LoRa (Long Range) communication protocol, which is known for its long-range coverage and low power consumption. Supporting this core functionality, various standard embedded communication protocols such as SPI, UART, and I2C are also implemented within the STM32F103C8T6 microcontroller environment to enable seamless interaction between modules.

LoRa, short for "Long Range," is a wireless modulation technique derived from chirp spread spectrum (CSS) technology. It allows data to be transmitted over long

distances—up to several kilometers—with minimal power usage, making it ideal for battery-powered IoT devices. In this project, the LoRa SX1276 module operating at 915 MHz is used for wireless communication between two nodes: a transmitter unit and a receiver unit. The transmitter reads data from a DHT11 temperature and humidity sensor and sends it wirelessly to the receiver. The receiver, upon successfully receiving the packet, displays the data on an OLED screen and optionally sends it to a PC via UART for logging or further processing.

LoRa provides several configurable parameters that affect performance, including spreading factor (SF), bandwidth, coding rate, and transmit power. These parameters are selected based on the range, power, and data rate requirements of the application. In this project, typical values such as SF7 or SF9 are chosen to balance between transmission speed and range. The LoRa protocol also includes features like cyclic redundancy check (CRC), preamble detection, and automatic frequency control, which significantly improve the reliability of communication in noisy environments.

The data transmitted over LoRa is structured into packets. Each packet includes fields such as a header, payload (sensor data), and checksum for validation. The STM32 microcontroller formats sensor readings (e.g., temperature and humidity) into a string or byte array and sends it over SPI to the LoRa module for transmission. On the receiver end, the same module captures the signal, decodes the packet, and passes it to the STM32, which extracts and parses the data before displaying it.

Serial Peripheral Interface: The communication between the STM32F103C8T6 microcontroller and the LoRa module is handled using the SPI (Serial Peripheral Interface) protocol. SPI is a synchronous serial communication protocol that provides high-speed data exchange between the master (STM32) and slave (LoRa SX1276). The SPI interface in this project uses four main lines: MOSI (PA7), MISO (PA6), SCK (PA5), and NSS/CS (PA4). The STM32 acts as the master device and initiates communication by setting the chip select (CS) line low. Data is then transferred in full-duplex mode—while the master sends data through MOSI, it simultaneously receives data via MISO.

The advantage of using SPI lies in its simplicity, speed, and full-duplex capability, which are particularly important when working with time-sensitive wireless modules like LoRa. The STM32 HAL or LL libraries are used to configure SPI parameters such as clock polarity, phase, bit order, and baud rate. These settings are carefully tuned to match the LoRa module's specifications and ensure smooth operation. SPI also enables fast, real-time transmission of control commands and payload data, making it a backbone protocol for this system's communication.

Serial Debugging and PC Interface: In addition to LoRa communication, the UART (Universal Asynchronous Receiver/Transmitter) interface is used in this project to establish serial communication between the STM32F103C8T6 and a computer or external terminal. UART is configured using PA9 (TX) and PA10 (RX) pins on the STM32. It plays a crucial role in debugging, monitoring, and logging system behavior during development and testing.

Using a USB-to-TTL converter, the UART connection allows the developer to observe real-time system data such as sensor values, LoRa transmission status, and error messages. This interface is invaluable for verifying whether data is being correctly sent and received, and it also aids in troubleshooting any runtime issues. The simplicity and reliability of UART make it an essential secondary communication channel that complements the primary LoRa wireless link.

OLED Display Communication: The I2C (Inter-Integrated Circuit) protocol is used for communication between the STM32 microcontroller and the 0.96-inch OLED display. I2C is a two-wire, synchronous communication protocol that operates with one master and one or more slave devices. In this project, the STM32 functions as the master, and the OLED display acts as the slave. The communication lines are SCL (PB6) for clock and SDA (PB7) for data.

The I2C interface is initialized using STM32CubeMX or manually via the HAL libraries. The OLED display is updated with the data received via LoRa, offering a real-time visualization of temperature and humidity readings. This interface is efficient and requires only two GPIOs, making it ideal for applications where pin count is a concern. Additionally, I2C allows for future scalability, such as adding multiple sensors or displays on the same bus using unique addresses.

CHAPTER 4: RESULTS AND DISCUSSION

The results and discussion section provides an analytical evaluation of the system's performance based on experimental observations, testing, and validation carried out during the implementation phase. The objective of this project was to develop a reliable, real-time, long-range wireless communication system using the STM32F103C8T6 microcontroller and LoRa SX1276 module, capable of collecting environmental sensor data and transmitting it effectively to a remote receiver unit. Through systematic testing, the system successfully demonstrated core functionalities including sensor data acquisition, data packet formation, wireless transmission, real-time reception, and display. The section below discusses the observed behaviour of each subsystem, data integrity, range performance, reliability, and potential areas for optimization.

System Functionality Verification: The system was first verified using fundamental test programs, such as LED blinking, UART communication, and sensor readings. These baseline tests ensured the microcontroller's GPIO, timer, and peripheral configuration were functioning correctly. Once the sensor (DHT11) was successfully interfaced, the transmitter module began collecting real-time temperature and humidity values. These values were formatted into structured messages and sent to the LoRa module via SPI.

The LoRa module, configured at 915 MHz with appropriate spreading factor and bandwidth settings, successfully transmitted the data to the receiver unit, which displayed the output on a 0.96-inch OLED screen via I2C. The receiver also forwarded the same data to a connected PC terminal through UART, allowing real-time monitoring on Tera Term. This confirmed the complete communication path—from environment sensing to wireless transmission, reception, and visualization—was functional and synchronized.

Data Transmission Integrity: During testing, the data packets transmitted over LoRa showed consistent integrity. Each packet sent from the transmitter was received accurately at the receiver, with the correct temperature and humidity readings displayed on the OLED. The system operated using unidirectional communication, without acknowledgment or retries, to maintain simplicity and lower power consumption. Nonetheless, there were minimal instances of packet loss, primarily due to environmental noise or temporary power instability. Even so, the system demonstrated high reliability within the designed communication range.

To ensure correct reception, the payload was formatted with delimiters or checksums. Debugging via UART confirmed that the sensor values were consistent with actual environmental readings, indicating correct sensor interfacing and data formatting. The

parsing mechanism at the receiver end was able to separate values and display them cleanly on the OLED screen, demonstrating robust handling of incoming data streams.

Communication Range and Performance: The LoRa SX1276 module was tested in both indoor and outdoor environments to evaluate its communication range. Indoors, the system maintained stable connectivity up to 50 meters, even through multiple walls and obstructions. In outdoor conditions with line-of-sight, reliable communication was observed up to 500–700 meters using a standard wire antenna. With optimized antenna design or placement, the range can be extended further. These observations validate LoRa's potential for wide-area monitoring in agricultural, industrial, or smart city applications where long-range wireless transmission is essential.

The power consumption of the system was also observed to be within acceptable limits. Since LoRa operates at low power and data was transmitted at fixed intervals, the system is well-suited for battery-powered or energy-harvesting applications. Sleep modes and power management features of the STM32 can be further implemented to optimize energy efficiency for continuous deployment.

User Interface and Usability: The integration of the OLED display significantly improved the usability of the system. It allowed real-time visualization of the received sensor data at the receiver end, eliminating the need for an external PC or serial monitor in standalone deployments. The display was clear, responsive, and updated without noticeable lag. Future improvements could include adding a graphical interface, historical data view, or integration with cloud platforms.

Additionally, the use of UART for debug output provided valuable insights during development and testing. It allowed the developer to print messages, monitor sensor output, and verify transmission behaviour. This feature proved essential for troubleshooting peripheral configurations and validating firmware logic.

Discussion and Opportunities for Improvement: While the system successfully achieved its intended objectives, there remain opportunities for enhancement. Implementing bidirectional communication could improve reliability by allowing acknowledgments and retransmissions in case of data loss. Incorporating encryption or data authentication would enhance security, especially for applications involving sensitive or critical data.

From a firmware perspective, introducing interrupt-driven communication and FreeRTOS-based task management would allow better scheduling, improve responsiveness, and enhance multitasking in future versions. On the hardware side, transitioning from a breadboard to a custom PCB would reduce noise, improve reliability, and provide a professional, compact form factor suitable for deployment.

Furthermore, expanding the system to support multiple sensor nodes and transmitting to a central receiver would demonstrate LoRa's full potential in creating scalable, mesh-like IoT networks. Data collected can be logged, visualized on dashboards, or transmitted to cloud services such as Thing Speak or Blynk using Wi-Fi or GSM extensions.

In conclusion, the implemented system performed successfully across all primary metrics: sensor integration, real-time wireless communication, data integrity, and visual display. The use of STM32F103C8T6 in conjunction with LoRa modules proved to be an effective combination for creating a low-cost, low-power, long-range sensor monitoring platform. The system was stable under various conditions, easy to use, and demonstrated potential for further enhancement. These positive outcomes validate the proposed approach and serve as a strong foundation for future development and deployment in real-world embedded and IoT applications.

4.1 Observations from Peripheral Testing

Peripheral testing is a vital step in validating the correct functioning and integration of microcontroller-based systems. In this project, the STM32F103C8T6 (Blue Pill) microcontroller served as the central unit, interfacing with various peripherals including the DHT11 sensor, LoRa SX1276 module, OLED display, and UART terminal. Each peripheral was tested individually and in combination to ensure proper initialization, stable operation, and reliable data exchange. The observations from these tests provide insight into the performance, responsiveness, and limitations of the system under realistic operating conditions.

GPIO and LED Control: The simplest test began with using GPIO to control an LED connected to PC13, a common onboard LED pin on the Blue Pill board. The LED blinked at regular intervals, confirming correct GPIO configuration and clock setup. Further testing showed that GPIO output was reliable even when toggled rapidly via timer interrupts. GPIO pins configured as input were tested using tactile switches; the logic level changes were read correctly with internal pull-up resistors enabled. This validated the microcontroller's ability to handle digital I/O operations required for user input and control signals (such as for LoRa module reset or interrupts).

DHT11 Sensor (Digital Input Testing): The DHT11 temperature and humidity sensor was connected to a GPIO pin (e.g., PB1) for one-wire digital data communication. During testing, the microcontroller successfully sent the start signal and received a 40-bit response from the sensor. Timing analysis revealed that accurate delays in the microsecond range were essential to capture the correct bits. Minor issues were initially observed when delays were implemented using standard software loops, causing

inconsistent readings. This was resolved by implementing hardware timer-based delays or using SysTick-based microsecond delay functions.

Once stabilized, the sensor output matched expected environmental conditions, and the values remained stable across consecutive readings. Error handling was added to ignore invalid readings or checksum mismatches. These tests confirmed the STM32's capability to manage precise timing and bit-level digital communication, especially important when dealing with non-standard communication protocols like that of the DHT11.

SPI Interface and LoRa Communication: The SPI interface was tested thoroughly as it served as the primary communication link between the STM32 and the LoRa SX1276 transceiver module. SPI lines (PA4–PA7) were configured for full-duplex master mode operation. During testing, initial communication failures were observed due to incorrect SPI clock polarity and phase settings, which were resolved after referring to the LoRa module's datasheet and matching CPOL and CPHA values accordingly.

Once configured properly, the LoRa module responded correctly to SPI commands such as register reads and writes. Transmit and receive operations were validated by sending dummy data between nodes and monitoring responses via UART and OLED. The system showed consistent SPI performance, with no data corruption or misalignment, even during continuous packet transmission. These results confirmed that the SPI bus was correctly initialized and stable under real-world communication conditions.

I2C Interface and OLED Display Testing: The I2C interface was used to drive a 0.96-inch OLED display, connected through PB6 (SCL) and PB7 (SDA). Peripheral initialization using STM32CubeMX allowed straightforward configuration of clock frequency and I2C mode. During testing, the OLED displayed characters and symbols correctly when initialized with proper driver libraries (such as u8g2 or Adafruit GFX compatible code). However, minor display flickering was observed during rapid refresh cycles, which was mitigated by introducing screen update delays and avoiding full-frame redraws unless necessary.

The display successfully updated with sensor data received via LoRa, verifying that the I2C interface could operate concurrently with SPI and UART without conflict. These observations demonstrated the STM32’s capacity to handle multiple I2C transactions with precise timing, even when other peripherals were actively engaged.

4.2Performance and Limitations

The performance of an embedded system is determined by how well it meets its functional requirements under real-world operating conditions. In this project, the primary performance objectives were to achieve reliable wireless communication, accurate sensor data acquisition, real-time data display, and low-power operation. These were implemented using the STM32F103C8T6 microcontroller, LoRa SX1276 transceiver, and supporting peripherals such as the DHT11 sensor, OLED display, and UART interface. Throughout the development and testing phases, the system demonstrated commendable results in terms of responsiveness, communication range, and stability. However, as with any prototype or early-stage deployment, certain limitations were observed that present opportunities for future enhancement.

System Performance: The system successfully achieved end-to-end data flow from the environmental sensor to the receiver display unit. The DHT11 sensor provided accurate temperature and humidity readings that were sampled at regular intervals. These readings were formatted by the STM32 and transmitted using the LoRa SX1276 module operating at 915 MHz, using an optimized configuration of spreading factor, coding rate, and bandwidth. The average transmission time was well within acceptable limits, ensuring that new data was consistently available to the receiver.

The LoRa communication showed excellent range in both indoor and outdoor environments. Indoors, the system maintained a stable connection across 50 meters through multiple walls. In an open outdoor field, the effective range extended up to 500–700 meters using simple wire antennas, with minimal packet loss. With proper antenna tuning and use of directional or high-gain antennas, this range can be extended even further, showcasing the system's potential for wide-area deployment in agriculture, smart cities, or remote monitoring.

The OLED display provided real-time visualization of the received data. Refresh rates were responsive, and screen updates occurred promptly after packet reception. This allowed users to view sensor readings in real-time without relying on external monitoring tools. In addition, the use of UART serial output enabled seamless logging and debugging, providing a useful fallback or parallel data channel. Overall, the STM32F103C8T6 efficiently handled all peripheral interactions, including SPI (LoRa), I2C (OLED), UART (debug), and GPIO (sensor and control), without noticeable performance degradation.

Sensor Accuracy and Resolution: The DHT11 sensor, while easy to use and low-cost, offers limited resolution and slower response times compared to more advanced alternatives like the DHT22 or BME280. Its ±2°C accuracy and limited range for humidity measurements may not be sufficient for applications requiring high precision.

Inconsistent readings were occasionally observed due to timing issues, especially during startup or in humid conditions.

Unidirectional Communication: The current implementation only supports one-way communication from transmitter to receiver. While this reduces complexity and power consumption, it limits the system's ability to handle acknowledgments (ACK), retransmissions, or remote reconfiguration. In high-reliability applications, bidirectional communication would be valuable to confirm successful reception and maintain system synchronization.

No Error Correction or Encryption: Although LoRa offers robust modulation schemes and error checking, the present design does not implement additional data validation techniques beyond basic string parsing or delimiters. Also, there is no data encryption, which could expose sensitive information if intercepted. In secure or mission-critical applications, adding AES encryption and CRC-based error checking would enhance system robustness and confidentiality.

Power Optimization: Power consumption was not fully optimized in this prototype. The STM32F103C8T6 was running continuously without utilizing sleep or low-power modes. Similarly, the LoRa module remained active at all times, which is not ideal for battery-powered or energy-harvesting applications. Power management techniques such as sleep scheduling, watchdog timers, and peripheral gating should be incorporated in future iterations.

Prototype Limitations: The system was built and tested on a breadboard, which introduces noise, loose connections, and potential instability due to floating signals or inconsistent grounding. For reliable deployment, the design should be transitioned to a custom PCB with proper trace routing, ground planes, and voltage filtering. This would also reduce the system's size and improve its mechanical durability.

Limited Data Visualization: The OLED display was effective for basic monitoring but limited in terms of graphical capability and historical data display. For enhanced usability, the system could be integrated with a mobile or web dashboard via Wi-Fi or GSM modules, enabling advanced visualization, data logging, and remote access.

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

The objective of this project was to design and implement a reliable, low-power, long-range embedded communication system using the STM32F103C8T6 microcontroller and the LoRa SX1276 module for the wireless transmission of sensor data. The system successfully integrated multiple core components—DHT11 temperature and humidity sensor, LoRa transceiver, OLED display, and serial communication interface—to form a complete transmitter-receiver solution. The transmitter node collected real-time environmental data, which was then formatted and transmitted wirelessly using the LoRa module. The receiver node received and parsed the data, displaying the output on an OLED screen and sending debug logs via UART.

Throughout the development and testing phases, the system demonstrated consistent and accurate performance. The use of the STM32F103C8T6, with its robust peripheral support for SPI, I2C, UART, and GPIO, proved ideal for handling multiple sensor interfaces and communication protocols. The LoRa SX1276 module delivered strong signal stability and long-range wireless transmission, achieving communication distances of up to 700 meters in open environments with minimal data loss. The OLED display provided a compact and effective way to present real-time data locally, while UART served as a critical interface for debugging and verification during the development process.

The firmware was developed using STM32CubeIDE, with all peripherals configured via STM32CubeMX. Debugging was efficiently performed using ST-Link V2 and a USB-to-TTL serial interface, helping to identify and resolve software or configuration-related issues. The system responded well under continuous operation and validated the intended design principles, including modularity, low-power operation, and long-range wireless capability.

This project successfully demonstrates how microcontroller-based systems can be combined with wireless technologies like LoRa to create effective Internet of Things (IoT) solutions. The integration of embedded peripherals, communication protocols, and basic user interfaces into a cohesive unit highlights the power and flexibility of modern embedded development platforms like the STM32 series.

5.2 Future Scope

While the system met its core objectives, there remains significant potential to enhance and expand the solution into a more robust, scalable, and feature-rich platform. The following points outline the future scope of this work and potential directions for continued development:

Bidirectional Communication: The current system supports unidirectional data transmission (transmitter to receiver only). In future implementations, adding bidirectional LoRa communication would allow the receiver to send acknowledgments (ACKs), control commands, or configuration parameters back to the transmitter. This would significantly improve reliability and enable real-time system reconfiguration.

Advanced Sensor Integration: The DHT11 sensor used in this project can be replaced with more accurate and sensitive sensors such as the BME280 or SHT31, which provide additional data like pressure and higher-resolution temperature and humidity values. Moreover, multiple sensors can be integrated to build a multi-node sensing network, enabling distributed data collection.

Cloud Connectivity: To enable remote data access and long-term storage, the receiver unit can be upgraded to support cloud integration via Wi-Fi (ESP8266/ESP32) or GSM/GPRS (SIM800) modules. This would allow sensor data to be sent to platforms like Thing Speak, Blynk, or custom dashboards for visualization, trend analysis, and user notifications.

Low-Power Optimization: Although LoRa is inherently low power, additional optimization strategies can be implemented such as deep sleep modes, interrupt-based wake-up, and duty-cycling. This would make the system more suitable for battery-powered applications and field deployments where energy efficiency is critical.

PCB Development: The current breadboard setup, while sufficient for prototyping, can be transitioned to a custom printed circuit board (PCB) for better mechanical stability, size optimization, and professional deployment. This would also allow enclosure design and improved EMI shielding.

Security Enhancements: As the system scales or handles sensitive data, implementing encryption protocols such as AES-128 and checksum validation becomes essential to protect against data interception or corruption. Secure key exchange mechanisms and firmware authentication can be introduced for industrial use cases.

Graphical User Interface (GUI): On the software side, a desktop or mobile GUI application can be developed to control, monitor, and visualize data from the receiver

node. This interface could include historical graphs, device control, and data export features, enhancing the usability of the system for non-technical users.

Multi-Node LoRa Network: The system can be expanded into a star or mesh network using LoRa, where multiple transmitters (nodes) send data to a central receiver or gateway. This architecture is ideal for large-scale monitoring scenarios such as precision agriculture, smart buildings, or environmental monitoring grids.

Final Remarks: This project lays the foundation for a fully functional, low-cost, long-range IoT solution built on open-source and industry-standard components. It serves as a scalable prototype for real-world applications in smart agriculture, industrial automation, weather stations, and environmental monitoring. By building upon the current implementation and incorporating the proposed future enhancements, the system can evolve into a versatile platform ready for commercial or academic deployment in modern wireless sensor network

5.3 Summary of Achievements

The development and implementation of the embedded communication system using the STM32F103C8T6 microcontroller and LoRa SX1276 transceiver has resulted in a successful demonstration of a low-cost, low-power, and long-range wireless sensor network. The project began with a conceptual goal of transmitting environmental sensor data wirelessly over long distances and culminated in a fully functional prototype capable of real-time data acquisition, wireless transmission, and user feedback through local display and serial output.

One of the key achievements of the project was the successful integration of multiple hardware components—DHT11 sensor, LoRa SX1276 module, 0.96" OLED display, and UART debugging interface—on a single embedded platform. The STM32F103C8T6 microcontroller served as a versatile control unit that managed data collection, processing, and communication using its built-in SPI, I2C, UART, and GPIO interfaces. All subsystems were tested and verified both independently and in conjunction, demonstrating high compatibility and reliable operation under real-world conditions.

The system achieved stable LoRa wireless communication with data successfully transmitted across distances of up to 700 meters in open environments, validating the effectiveness of LoRa modulation in long-range, low-power applications. The sensor data was consistently transmitted from the remote transmitter node to the receiver, with minimal packet loss and high reliability. These results confirm the viability of LoRa as a core technology for wireless monitoring and control systems.

In addition, the project showcased the use of STM32CubeIDE and STM32CubeMX for embedded firmware development, allowing structured and maintainable code generation. The use of ST-Link V2 for programming and in-circuit debugging, along with USB-to-TTL serial communication for real-time monitoring, streamlined the development process and facilitated efficient troubleshooting. The successful implementation of these tools reflects a deep understanding of the embedded development workflow and hardware-software integration techniques.

Another major accomplishment was the inclusion of an interactive user interface using an OLED screen, enabling real-time display of received sensor data without dependence on external systems. This enhanced the practicality and portability of the system, making it suitable for standalone deployments in field conditions.

The entire system was built and tested on a breadboard-based prototype, which provided the flexibility needed for iterative testing and component-level debugging. Despite being a prototype, the system maintained robustness and demonstrated resilience during continuous operation, further proving its suitability for real-world deployment scenarios.

Furthermore, the modular structure of the firmware and hardware layout makes the current system easily expandable. Future enhancements such as advanced sensors, cloud integration, security features, and multi-node support can be seamlessly added, thanks to the scalable architecture established during the initial development.

5.4 Recommendations and Enhancements

While the current implementation of the embedded communication system has achieved its fundamental objectives—accurate sensing, reliable long-range wireless transmission, and real-time data visualization—there remain several areas where enhancements can significantly improve functionality, scalability, efficiency, and usability. These recommendations are based on observed limitations, evolving technological possibilities, and best practices in embedded systems and IoT design. Implementing these recommendations will strengthen the project's robustness and bring it closer to production-level standards suitable for real-world applications.

Upgrade to Higher-Precision Sensors: The system currently utilizes a DHT11 sensor for measuring temperature and humidity. While it is adequate for basic testing and demonstrations, its limitations include relatively low accuracy, slow response time, and narrow humidity range. It is recommended to upgrade to sensors such as the DHT22, BME280, or SHT31, which offer higher resolution, better stability, and support for

additional environmental parameters such as barometric pressure and altitude. Using more advanced sensors will expand the application scope and improve data quality, especially in professional monitoring systems.

Implement Bidirectional Communication: Currently, the system supports unidirectional data flow from transmitter to receiver. For enhanced reliability and control, bidirectional communication should be implemented using LoRa's built-in two-way messaging capabilities. This would allow the receiver to send acknowledgments (ACKs), configuration commands, or error messages back to the transmitter. Such feedback can help detect and recover from transmission failures, optimize data transmission intervals, and dynamically adjust sensor sampling rates or LoRa parameters based on environmental conditions or network traffic.

Incorporate Power Optimization Strategies: One of the major strengths of LoRa technology is its potential for ultra-low power operation. To fully leverage this in long-term or remote deployments, it is advisable to implement power-saving features such as: Sleep modes for the STM32 and LoRa modules between transmission cycles, Interrupt-based wakeup from timers or sensor events. Dynamic peripheral gating to disable unused modules during idle periods.
These enhancements can extend battery life significantly, making the system viable for off-grid applications powered by small batteries or solar panels.

Transition to PCB Design: The current system is built on a breadboard, which, although useful for prototyping, introduces challenges such as signal noise, mechanical instability, and higher failure rates due to loose connections. For greater reliability, it is strongly recommended to migrate the design to a custom Printed Circuit Board (PCB). This would allow for: Optimized routing of high-speed communication lines (SPI, I2C), Proper power supply decoupling, Compact, enclosure-ready form factors, Long-term durability and professional deployment readiness.

Improve Data Security and Integrity: As IoT systems grow in complexity and connectivity, data security becomes a critical concern. The current system does not include any form of encryption or error correction beyond basic string parsing. It is recommended to implement: AES encryption for secure transmission over LoRa. CRC checks or checksums to verify data integrity. Access control or key management strategies for authenticated communication.
These features will be especially important in applications involving sensitive data (e.g., health, safety, agriculture) or multi-node systems operating in public or industrial environments.

Add Cloud and Mobile Integration: To enhance accessibility and scalability, the system can be extended to send data to cloud platforms such as ThingSpeak, Firebase, or AWS

IoT. This can be accomplished by interfacing the STM32 or the receiver node with a Wi-Fi (ESP8266/ESP32) or GSM/GPRS (SIM800) module. Cloud integration would enable: Historical data logging. Remote system monitoring. Graphical dashboards and mobile notifications.
Incorporating cloud services also opens the door to advanced analytics, machine learning, and decision-making capabilities.

Support for Multi-Node Networks: Expanding the system to handle multiple transmitter nodes communicating with a single receiver or central gateway is a natural next step. This enhancement would demonstrate LoRa's strength in enabling large-scale, decentralized wireless networks. Proper addressing, time-division multiplexing, and queue management would need to be implemented in software. This type of architecture is ideal for smart farming, environmental monitoring, and smart city applications.

Enhance the User Interface: The OLED screen used in the current prototype provides basic real-time data display. However, it is recommended to enhance the user interface (UI) with features such as: Menu navigation via buttons or rotary encoders. Historical data graphs.Visual alerts for threshold breaches (e.g., over-temperature). Display of system health or network signal strength. This would increase system usability and provide a more intuitive experience for end users in standalone deployments.

Modular Firmware Design: To ensure long-term maintainability and adaptability of the software, the firmware should be refactored into modular code blocks for each subsystem (sensor, LoRa, display, UART, etc.). Using a Real-Time Operating System (RTOS) such as FreeRTOS would further allow for prioritized task scheduling, non-blocking execution, and inter-task communication. This makes the system more scalable and suitable for complex, multi-functional applications.

Conclusion of Recommendations: By implementing these enhancements, the current prototype can evolve into a sophisticated, field-deployable IoT solution. These recommendations will significantly expand the system's functionality, reliability, and practical use cases, ranging from basic educational demonstrations to full-scale environmental monitoring systems. They also provide a clear pathway for continued research, innovation, and potential commercialization.

5.5 REFERENCE

1.D. H. Kim, J. Y. Lim and J. D. Kim,“Low-Power, Long-Range, High-Data Transmission Using Wi-Fi and LoRa”, 6th International Conference on IT Convergence and Security (ICITCS), Prague, Czech Republic, 2016, pp. 1-3,
DOI: 10.1109/ICITCS.2016.7740351.

2. U. Noreen, A. Bounceur and L. Clavier, “A study of LoRa low power and wide area network technology”, International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Fez, Morocco,pp. 1-6,
DOI: 10.1109/ATSIP.2017.8075570.

3.Sanchez-Iborra, R., G. Liaño, I., Simoes, C., Couñago, E., & Skarmeta, A. F. . ,“Tracking and monitoring system based on LoRa technology for lightweight boats”.December 2018.Electronics 8(1):15

DOI: 10.3390/electronics8010015LicenseCC BY 4.0

4.Ziqi Lin, Xu Zhang, Shimin Gong, Lanhua Li, Zhou Su, Bo Gu,“Matching-Driven Deep Reinforcement Learning for Energy-Efficient Transmission Parameter Allocation in Multi-Gateway LoRa Networks”.From: Xu Zhang [view email]
[v1] Sat, 16 Sep 2023 11:37:23 UTC (2,953 KB)

5.Ziqi Lin, Xu Zhang, Shimin Gong, Lanhua Li, Zhou Su, Bo Gu,“Matching-Driven Deep Reinforcement Learning for Energy-Efficient Transmission Parameter Allocation in Multi-Gateway LoRa Networks”.From: Ziqi Lin [view email]
[v1] Thu, 18 Jul 2024 00:54:26 UTC (2,126 KB)

6.Ganghui Lin, Ahmed Elzanaty, Mohamed-Slim Alouini,“LoRa Backscatter Communications: Temporal, Spectral, and Error Performance Analysis”.From: Ganghui Lin [view email] [v1] Sun, 4 Jun 2023 10:30:04 UTC (8,299 KB)
[v2] Tue, 20 Jun 2023 14:33:44 UTC (8,323 KB)

7.Ryotai Airiyoshi, Mikio Hasegawa, Tomoaki Ohtsuki, Aohan Li,“Energy Efficient Transmission Parameters Selection Method Using Reinforcement Learning in Distributed LoRa Networks)”.From: Aohan Li [view email]
[v1] Tue, 15 Oct 2024 04:49:56 UTC (2,355 KB)
[v2] Wed, 22 Jan 2025 04:25:10 UTC (2,354 KB)

8.Jesse Duran;Yiyan Li,“An Ultra Low Power LoRa Mesh for Low Cost and Real Time Resort Pool Temperature Monitoring”.DOI:10.1109/ICETCI57876.2023.10177034

9.Eynar Calle Viles; Edgar Roberto Ramos Silvestre; Elias Brayan Choque Maydana; Wilder Orellana Lopez; Cristian Capriles Olivera,“Performance Assessment of a Low Power Wide Area wireless Networks based on Lora Technology using IOT devices in different Bolivia Areas ”.DOI:10.1109/ETCM63562.2024.10746217

10.Diyuan Shen;Ling Huang;Xiaobo Zhang;Zhangqin Huang,“Research on ReaL Time Data Wireless Communication System Based on LoRa Technology”.DOI:10.1109/ICCCS61882.2024.10602933