OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

Autonomous Distributed Node-Based Rail Control with Fault-Tolerant Architecture

K M Yadukrishnan , Nithin Kumar S Lavanya Vaishnavi D

[1]*UG scholar, Dept. of Electronics and Communication Engineering, R L Jalappa institute of Technology, Doddaballapur, Affiliated to VTU Belagavi. India*

[a] kmyadukrishnan92@gmail.com , [b]nithinkumars7777@gmail.com

[2]*Assistant Professor, Department of Electronics and Communication Engineering*
R L Jalappa Institute of Technology, Doddaballapur, Affiliated to VTU Belagavi, India
lavanyavaishnavi@gmail.com

Abstract: ***The research article presents a Distributed Rail Signalling and Traffic Control System (DRiSTaCS) that replaces traditional centralized signalling with a fully distributed, node-based architecture for modern rail networks. Each train, signal, and switch functions as an autonomous node, processing local sensor data and executing real-time control logic over encrypted TCP/IP links within a closed network. The system integrates both fixed block signalling in dense traffic corridors and moving block signalling on long-distance routes, enabling dynamic headway adjustment while preserving safety margins. Multi-modal sensor arrays, including magnetic, optical, and pressure sensors, continuously verify track and switch states and feed predictive maintenance algorithms that transition asset management from reactive to proactive strategies. The communication backbone operates in the 850 MHz to 2.1 GHz band with redundant GSM/GPRS channels, ensuring low-latency data exchange and resilience against link failures and cyber intrusions. A self-healing design allows neighboring nodes to reroute traffic and assume control during localized faults, avoiding single points of failure inherent in legacy architectures. Hardware is modular, combining Linux-based high-performance controllers, RP2040 and STM32 microcontrollers, and dedicated ultra-low-power boards optimized for deterministic timing and continuous field deployment. Middleware in C/C++ and Node.js coordinates protocol handling and data aggregation, while computationally intensive tasks such as sensor fusion and route optimization are offloaded to specialized processing nodes. Experimental results on custom communication modules and differential high-speed buses demonstrate reliable operation up to the sub- MHz range with robust packet delivery and timing integrity. Overall, DRiSTaCS delivers a scalable, secure, and automation-centric signalling framework that improves safety, availability, and maintainability in next-generation railway systems.***

Keywords: block signalling, communication security, distributed architecture, DRiSTaCS, fault tolerance, predictive maintenance, railway automation, real-time control, sensor fusion, traffic management

---

I. INTRODUCTION

Railway transportation systems form the backbone of modern mobility, enabling the efficient movement of passengers and freight across vast distances. However, legacy signalling and traffic control infrastructures continue to pose significant challenges to operational efficiency, safety, and scalability. Traditional centralized architectures, while historically effective, are increasingly inadequate in addressing the demands of high-density traffic, real-time data processing, and integration with emerging technologies such as IoT sensors and predictive analytics. These limitations underscore the urgent need for innovative solutions that can transform railway operations into resilient, adaptive, and future-ready networks.

The Distributed Rail Signalling and Traffic Control System (DRiSTaCS) represents a paradigm shift in railway signalling technology. Unlike conventional centralized systems that rely on a single control hub, DRiSTaCS disperses decision-making authority across multiple interconnected nodes. Each node operates autonomously, processing local sensor data and executing control algorithms in real time. This distributed architecture eliminates the single-point-of-failure risk inherent in centralized systems, while simultaneously enhancing fault tolerance and scalability. By integrating both fixed block and moving block signalling paradigms, DRiSTaCS achieves a hybrid design that balances predictability in high-density environments with dynamic adaptability on long-range routes.

At the core of DRiSTaCS lies a robust communication infrastructure built on encrypted TCP/IP protocols. This closed network design ensures secure, low-latency data exchange among trains, trackside signals, and control units. Supplementary communication modules such as GSM/GPRS provide redundancy, guaranteeing that critical alerts are transmitted even under adverse conditions. The system's ability to maintain deterministic real-time performance through customized firmware further

strengthens its reliability. By restricting access to authenticated devices only, DRiSTaCS minimizes vulnerabilities and mitigates cybersecurity risks, a growing concern in modern rail networks.

Sensor integration is another cornerstone of the system. Trackside switches, signals, and onboard train computers are equipped with multi-modal sensor arrays, including magnetic, optical, and pressure sensors. These sensors provide continuous verification of rail conditions and switch positions, feeding data into predictive maintenance modules. Over time, the system leverages machine learning algorithms to forecast component degradation and schedule maintenance proactively, reducing downtime and operational costs. This transition from reactive to predictive maintenance marks a significant advancement in railway asset management.

From an operational perspective, DRiSTaCS emphasizes automation and minimal human intervention. Its self-healing network architecture allows nearby nodes to reroute communications and assume control responsibilities in the event of a failure. In critical scenarios such as emergency braking, train-to-train communication protocols enable rapid cascades of encrypted messages, ensuring instantaneous speed adjustments across the network. Such distributed interventions outperform legacy systems that rely on slower centralized commands, thereby enhancing safety and reliability.

The software architecture of DRiSTaCS complements its hardware design. High-performance nodes running Linux handle complex routing and switching logic, while low-power microcontrollers such as RP2040 execute time-critical tasks in C for minimal latency. Middleware layers in C++ and Node.js orchestrate real-time data parsing and synchronization, while computationally intensive processes like sensor fusion and route optimization are offloaded to dedicated nodes running Julia. This layered approach ensures responsiveness under heavy data loads and seamless interoperability across heterogeneous devices.

In summary, DRiSTaCS addresses the pressing challenges of modern railway systems by combining distributed control, secure communication, sensor fusion, and predictive maintenance into a unified framework. Its objectives—real-time monitoring, scalability, and automation—align with the evolving demands of global rail networks. By integrating advanced signalling technologies such as CBTC and ETCS within its architecture, DRiSTaCS not only enhances operational efficiency but also sets the stage for future innovations in railway safety and reliability.

II.LITERATURE SURVEY

J. Luo et. al., proposes underwater wireless sensor networks (UWSNs) face unique propagation challenges in water, prompting the development of novel routing protocols that primarily use acoustic and optical signals rather than RF, which attenuates severely. Recent trends incorporate machine learning (ML) techniques like multi-agent reinforcement learning and Q-learning for energy-efficient routing. The surveyed paper categorizes protocols into energy-based, data-based, and geographic-based methods. Modeling employs simulation frameworks replicating underwater conditions, including high delays, multipath fading, noise, and limited bandwidth. Simulations cover 2D static networks and 3D setups with mobile AUVs to mirror real scenarios. Key numerical results highlight performance gains: energy-based protocols extend network lifetime by 20–30%; geographic protocols achieve >85% packet delivery ratios; ML approaches like Q-learning reduce end-to-end delays by ~15% while balancing node energy use, preventing early failures. These protocols support vital applications in ocean monitoring, seismic sensing, environmental assessment, and seabed exploration, ensuring reliable data amid Doppler shifts, currents, and harsh conditions. Future directions include adaptive ML, energy harvesting, and cross-layer designs.

Y. Si et. al. proposes a multi-layer communication architecture for a smart micro-grid testbed, featuring customized Edge Intelligent Devices (EIDs) with SPI and Modbus TCP/IP protocols. A key innovation is dual-core DSPs per DER inverter—one for system control (e.g., MPPT and grid-forming) and one for communication—enhancing data efficiency and minimizing interference. The architecture spans four layers: Process Layer (DER inverters with dual DSPs), Interface Layer (Raspberry Pi for SPI-based signal translation), Sub-station Layer (EID for data aggregation via Modbus TCP/IP and wireless links across AEPS), and Supervisory Layer for oversight. Hardware validation confirms superior performance: dual DSPs boost communication speed and cut control-communication delays by ~40%; SPI reaches 12.5 MHz for rapid inverter-RPi feedback; Modbus TCP/IP minimizes latency for substation integration, supporting real-time micro-grid operations. Applicable to PV-heavy smart grids with storage, the framework enables seamless DER coordination. Future work may integrate IEC-61850 for interoperability and AI analytics via e-SEOP for energy forecasting and stability.

X. Luo et al., This paper proposes a lightweight, privacy-preserving protocol for heterogeneous IoT using symmetric key encryption via Logistic Map-based chaotic systems for key generation, avoiding resource-heavy asymmetric methods like RSA. Initialization involves pre-configured parameters for network joining, with the control center assigning dynamic session keys for secure D2D communication and real-time updates against replay attacks.

Experiments show ~40% faster encryption than RSA, with low computational overhead suitable for low-power devices. Future work could integrate AI for adaptive security, quantum resistance, while addressing key synchronization and side-channel vulnerabilities in large-scale IoT.

S. Babu et. al., introduces a Medium-Term Disruption Tolerant Software Defined Network (MDT-SDN), designed to handle link disruptions lasting 10 seconds to 6 minutes in next-generation wireless networks. Traditional approaches like Delay/Disruption Tolerant Networking (DTN), which rely on bundle protocols, are unsuitable due to excessive overhead. Instead, MDT-SDN utilizes SDN-controlled buffering at intermediate nodes, preventing unnecessary packet loss and retransmissions. A key innovation is the introduction of the STORE action in OpenFlow switches,

enabling temporary packet buffering within the existing TCP/IP framework during network disruptions. Additionally, the framework employs temporal graph modeling to optimize network routing. [4] MDT-SDN is highly applicable in wireless mesh networks, campus networks, disaster recovery communications, and IoT-based sensor networks, where network disruptions due to mobility or interference are frequent. Future enhancements may include: ntegration with 5G Networks: MDT-SDN can be extended to mobile edge computing architectures, improving service reliability in ultra-low-latency environments. Enhanced AI-Based Traffic Prediction: Machine learning models could replace Markov-based predictions, enabling real-time adaptation to network conditions. [4]

However, challenges remain, such as ensuring scalability for large-scale deployments, preventing excessive buffering delays in high-density networks, and maintaining controller responsiveness under variable mobility conditions. [4]

This research proposes an innovative SDN-based disruption-tolerant architecture, optimizing medium-term packet buffering and adaptive routing mechanisms in wireless networks. Let me know if you'd like any refinements or additional insights! [4]

W. Anani et. al., Proposed paper introduces a lightweight, secure wireless meter-bus (wM-Bus) protocol, specifically designed for energy-sensitive IoT devices such as smart meters. Unlike conventional security solutions that heavily rely on Transport Layer Security (TLS), the proposed protocol leverages the Noise Protocol Framework (NPF) to significantly reduce computational and power requirements. The innovative XX and NX handshake patterns within NPF enhance security without compromising performance. Additionally, the protocol eliminates static key transmission—reducing exposure to potential interception attacks. [5]

The research methodology follows a structured implementation spanning five distinct phases, covering protocol evaluation, integration, and security optimization. [5]

H. Li et al., proposed paper introduces a dual module parallel readout system designed for the HEPS-BPIX detector, leveraging 10 Gb TCP/IP transmission to significantly enhance data throughput. A key innovation is the dual modules readout board (DMRB), which integrates Field Programmable Gate Arrays (FPGAs) to manage high-speed data acquisition. The Beijing PIXel (BPIX) chips enable zero dead-time operation, ensuring continuous data collection. The 10 Gb TCP/IP firmware improves bandwidth efficiency, surpassing traditional UDP-based transmission methods. Additionally, the through silicon via (TSV) module enhances connectivity and scalability, allowing seamless integration with multiple detector modules. The high-speed optical fiber transmission and parallel architecture further optimize performance, making the system suitable for high-frame-rate imaging applications. [6]

III.PROBLEM STATEMENT

Traditional centralized railway signalling systems suffer from single-point failures, high latency in decision-making, and limited scalability amid rising traffic densities and integration demands for IoT and predictive analytics. These legacy architectures struggle with real-time data processing, vulnerability to cyber threats, and reactive maintenance, leading to frequent disruptions, safety risks, and escalating costs. Fixed block signalling constrains capacity on busy routes, while lacking hybrid adaptability for mixed urban-rural operations. Communication silos and human-dependent interventions further amplify delays during faults or emergencies. DRiSTaCS addresses these by deploying a fully distributed, self-healing node network that ensures resilient, automated control across heterogeneous rail environments.

IV.OBJECTIVES

To develop a distributed architecture replacing centralized control with autonomous nodes for trains, signals, and switches, enabling real-time local processing and eliminating single-point vulnerabilities.

Integrate hybrid fixed-moving block signalling with multi-modal sensors and encrypted TCP/IP communications to optimize capacity, safety, and predictive maintenance in diverse rail corridors.

Implement self-healing mechanisms and redundant channels for fault tolerance, ensuring uninterrupted operations, low-latency emergency responses, and scalable deployment via modular hardware-software design.

Methodology of Implementation

The Distributed Rail Signalling and Traffic Control System (DRiSTaCS) is engineered to revolutionize railway operations through a hybrid environment that harmonizes fixed and dynamic signalling methodologies. At the core of this system lies a decentralized network architecture where every entity—from trains and trackside signals to switching units—operates as an autonomous node. This decentralized approach not only eliminates the vulnerabilities of centralized, single-point control systems but also enables real-time decision-making and fault tolerance that are critical in safety-sensitive rail networks.

DRiSTaCS integrates two primary signalling techniques. In the zone where rail traffic is dense, the system employs Fixed Block Signalling. Here, the railway is divided into predetermined segments, usually spanning approximately two kilometres. Within each segment, a group of four to five interlinked signals collaborates to manage train entry and exit, monitor occupancy, and maintain synchronous control among adjacent blocks. In contrast, the system activates Moving Block Signalling on less congested mainline routes. In this dynamic mode, continuous real-time data regarding a train's speed, position, and track gradient are exchanged among vehicles, enabling the calculation of safe braking distances and allowing the adjustment of spacing between trains on the fly. The wireless communication between network nodes is facilitated by an encrypted TCP/IP stack, operating over a spectrum between 850 MHz and 2.1 GHz, ensuring both long-range effectiveness and high throughput.

Security and reliability are paramount in DRiSTaCS. The entire

system is built upon a closed network architecture where only authenticated devices can participate. Each node communicates using a robust, packet-based protocol with built-in error detection and recovery, ensuring high data integrity even under heavy load conditions. Redundant communication channels, such as low-power GSM/GPRS modules, are integrated to serve as backups in contingencies, thereby maintaining uninterrupted operations during network interruptions.

A distributed network design, devoid of any central controller, characterizes DRiSTaCS. Instead of relying on a single control room, each train, signal, and switch processes local sensor data autonomously and makes real-time operational decisions. This architecture significantly reduces latency during critical scenarios. In practice, station masters assume a strictly localized role, handling only intra-station switching operations, while the dynamic management of mainline routing is delegated to onboard systems that continuously interact with adjacent nodes. Emergency communication protocols—especially during sudden braking events—are executed in a peer-to-peer manner, allowing for a rapid cascade of alerts between cars, which in turn enables instantaneous adjustment of speeds and braking distances.

Switching and sensor integrations further exemplify the technical robustness of DRiSTaCS. Each track switch is equipped with multiple sensor modalities, including magnetic, optical, and pressure sensors, that help confirm proper alignment and rail contact. This multi-sensor verification process is critical in ensuring that switching operations are executed accurately; a switch is activated only after all sensors have conveyed a positive signal. In addition, these sensor networks feed real-time data to a maintenance and diagnostic system that continuously monitors equipment health. Predictive maintenance algorithms analyze historical sensor data to schedule interventions before failures occur, thereby reducing operational downtime and maintenance costs.

Inter-vehicular communications in DRiSTaCS are equally sophisticated. Trains are enabled to exchange continuous streams of operational data — from speed and location to directional vectors — through encrypted messages distributed over the wireless network. This train-to-train communication ensures a high degree of coordination, allowing the system to anticipate and avoid potential collisions. For instance, in emergency braking scenarios, an immediate and concurrent alert is shared with all following trains, triggering a coordinated reduction in speed and activation of emergency protocols to prevent rear-end collisions.

The overall reliability of the system is further enhanced by its self-healing capabilities. In the event of a fault or failure at any node, adjacent nodes automatically reroute communication channels and assume control of the affected area. In fixed block zones, if one signal fails, the remaining signals within the block collaboratively maintain traffic control, ensuring continuous network functionality despite localized failures. This resilience is complemented by integrated diagnostics that log every operational parameter, allowing maintenance teams to address potential issues proactively.

Block Diagram of Implementation

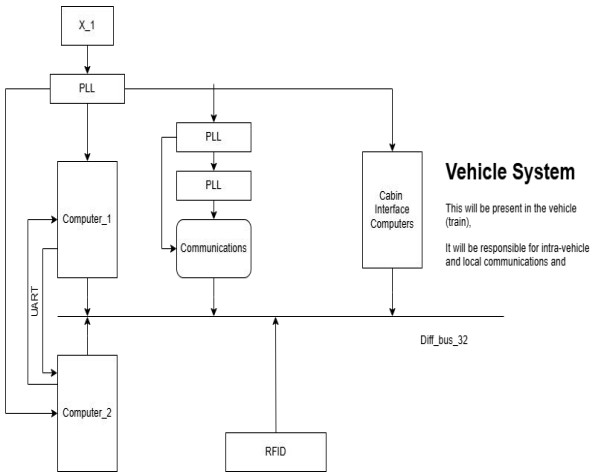

Figure : Vehicle System Architecture

Computer

Embedded computers, unlike general-purpose PCs, are purpose-built systems optimizing for specific functionalities. Their architecture typically features a microcontroller (MCU) or microprocessor (MPU) as the central processing unit (CPU), often integrating peripherals like Analog-to-Digital Converters (ADCs), Digital-to-Analog Converters (DACs), Timers, and various communication interfaces (UART, SPI, I2C, CAN) onto a single System-on-Chip (SoC). Memory includes Flash for program storage (firmware), SRAM for volatile data, and sometimes EEPROM for persistent configuration. Real-time operating systems (RTOS) manage task scheduling, ensuring deterministic behavior and low latency crucial for control applications. Power consumption, footprint, and cost are critical design constraints, often leading to specialized instruction sets (e.g., ARM Cortex-M) and custom hardware acceleration.

PLL

A Phase-Locked Loop (PLL) is a feedback control system that automatically adjusts the phase of a locally generated signal to match the phase of an input reference signal. Its core function is to synchronize the output signal's frequency and phase with the reference, even if the reference signal experiences variations or noise. This synchronization is achieved through a continuous comparison of the two signals, generating an error signal that drives a voltage-controlled oscillator (VCO) to adjust its output until phase and frequency lock are achieved.

The fundamental components of a PLL include a phase detector (PD), a low-pass filter (LPF), and a voltage-controlled oscillator (VCO). The phase detector compares the phase difference between the reference input and the VCO's output, producing a voltage proportional to this difference. This error voltage then passes through the low-pass filter, which smooths out high-frequency noise and provides a stable control voltage to the VCO. The VCO, in turn, generates an output signal whose frequency is directly controlled by this filtered voltage. This closed-loop feedback mechanism ensures that the VCO's output frequency and phase eventually converge with that of the input reference signal, establishing a "locked" condition. PLLs are integral to various

modern electronic systems, from frequency synthesis and clock generation in microprocessors to data recovery and demodulation in communication systems.

Crystal Oscillator

A crystal oscillator is an electronic circuit that uses the mechanical resonance of a vibrating crystal to generate a precise frequency signal. Quartz crystals are commonly used due to their stability and accuracy. These oscillators are essential in maintaining consistent timing in various electronic devices.

The principle behind crystal oscillators relies on the piezoelectric effect, where the crystal deforms when subjected to an electric field. This deformation produces vibrations, leading to a stable oscillation at a specific frequency. The frequency depends on the physical dimensions and cut of the crystal.

Crystal oscillators are widely used in applications where precise timing is crucial. They serve as clock signals in microprocessors, watches, radios, and communication systems. Their ability to maintain stability even under temperature variations makes them ideal for industrial and scientific applications.

The advantage of a crystal oscillator over other timing sources lies in its minimal frequency drift. This ensures reliable performance over extended periods. Additionally, crystal oscillators are resistant to external noise, enhancing signal clarity.

Different types of crystal oscillators exist, including Pierce, Colpitts, and Clapp configurations. Each type caters to specific needs, ensuring optimized performance across various industries. Their reliability makes them indispensable in modern electronics, ensuring synchronized operations in countless devices.

Computing Module

This schematic represents Compute Module shown in Fig 3.2 built around the RP2040 microcontroller, designed to provide a complete environment for programming, debugging, and interfacing with external hardware. At the centre is the RP2040 itself, which exposes its GPIO pins through header connectors, allowing flexible use for digital I/O, communication protocols, and peripheral control. The microcontroller relies on an external 12 MHz crystal oscillator with supporting capacitors and a resistor to provide a stable clock source, ensuring accurate timing for its dual-core operation. Power is managed through a regulator circuit that converts the USB supply into a clean 3.3 V rail, with decoupling capacitors placed strategically to stabilize voltage and reduce noise across the system.

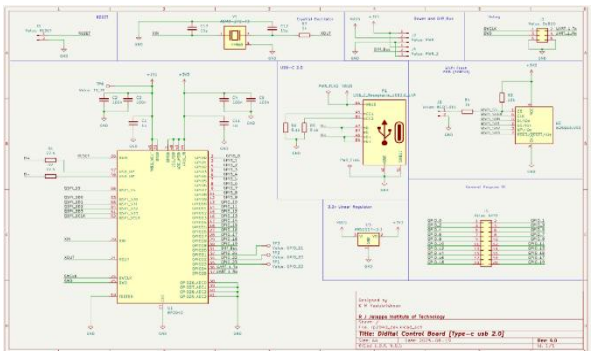

Figure 4.2: Compute Module Circuit

The USB interface serves as both a power input and a communication channel, with resistors and capacitors ensuring proper signal integrity. A reset button and a boot button are included to simplify firmware uploading and recovery, directly tied to the RP2040's control pins.

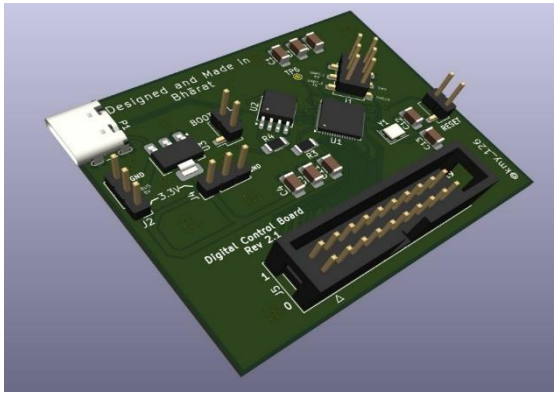

Figure 4.3: Compute Module 3D

For debugging, a dedicated SWD header provides access to low-level programming and troubleshooting. External flash memory connected via SPI stores the firmware, enabling the microcontroller to boot and run user applications. Together, these elements form a compact yet versatile board suitable for embedded development and experimentation.

COMS Module

This circuit as per Fig 3.4 represents a communications module that operates entirely with discrete components and supporting blocks, without relying on any microcontroller. The design begins with a regulated 3.3 V supply derived from a linear regulator, protected by a diode and stabilized with capacitors to ensure clean power delivery to the rest of the system. The transmitter stage is built using a pair of NPN transistors configured with biasing resistors and coupling capacitors, which together amplify and shape the outgoing signal. A similar transistor-based section is dedicated to generating a low-power PWM waveform, again relying on passive timing elements and transistor switching rather than digital control.

The signals produced by these stages are passed through a low-pass filter and attenuator network, which smooths the waveform and limits its bandwidth, producing a cleaner analog output suitable for communication purposes. Test points are distributed across the schematic, allowing engineers to probe the transmitter output, PWM generator, and filtered signals for analysis and tuning. External headers provide connections for power input and PC-side communication lines, enabling the circuit to be integrated into larger systems or tested in isolation. By avoiding microcontrollers, this design highlights a purely hardware-driven approach to signal generation and conditioning, relying on analog techniques and discrete logic for communication functionality.

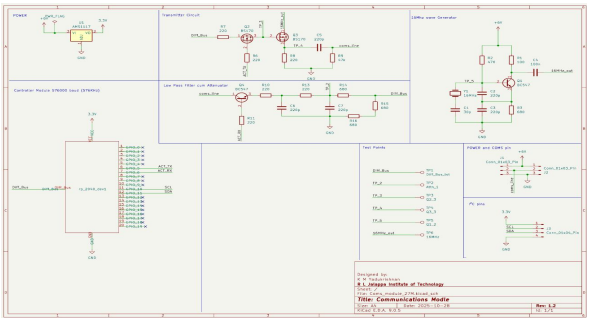

Figure 4.4 COMS Module Circuit

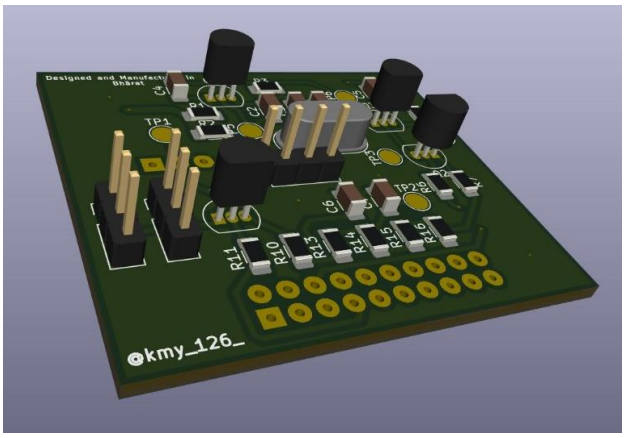

Figure 4.5: COMS Module 3D

Ultra Low Power Module

This schematic in Fig 3.6 represents a Ultra Low Power board designed around the STM32C011F4 microcontroller, providing the essential building blocks required for embedded system prototyping and experimentation. The circuit begins with a power regulation stage, where a +6 V input is stepped down to a stable 3.3 V supply using an AMS1117-3.3 linear regulator. Decoupling capacitors are placed at the regulator's output and near the microcontroller's supply pins to filter noise and ensure reliable operation.

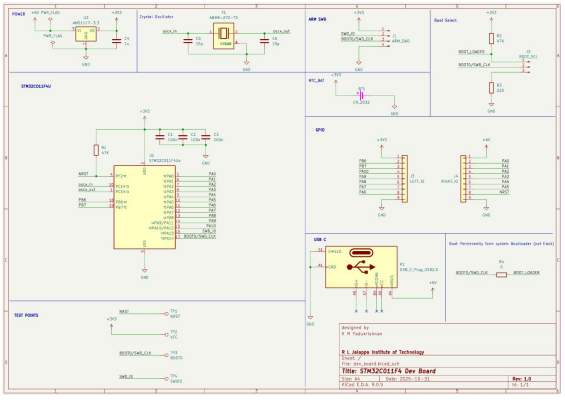

Figure 4.6: Ultra Low Power Module Circuit

The regulated 3.3 V rail powers the STM32 device as well as other supporting components. A crystal oscillator, along with its load capacitors, provides the microcontroller with a precise external clock source, ensuring accurate timing for system functions. The reset circuitry and boot configuration resistors allow the device to be placed into different modes, such as normal execution or bootloader mode, simplifying firmware programming and recovery.

The STM32C011F4U microcontroller is the central element of the design, with its GPIO pins broken out to headers for easy interfacing with external peripherals. These pins are grouped and labeled, making it straightforward to connect sensors, actuators, or communication modules. The board also includes an ARM SWD (Serial Wire Debug) connector, which provides a direct interface for programming and debugging the microcontroller using standard tools. To support real-time clock functionality, a coin-cell battery holder is included, allowing the RTC to keep time even when the main supply is disconnected.

Additional passive components such as pull-up resistors, filtering capacitors, and test points are distributed across the schematic to improve stability, simplify troubleshooting, and provide convenient access to critical signals.

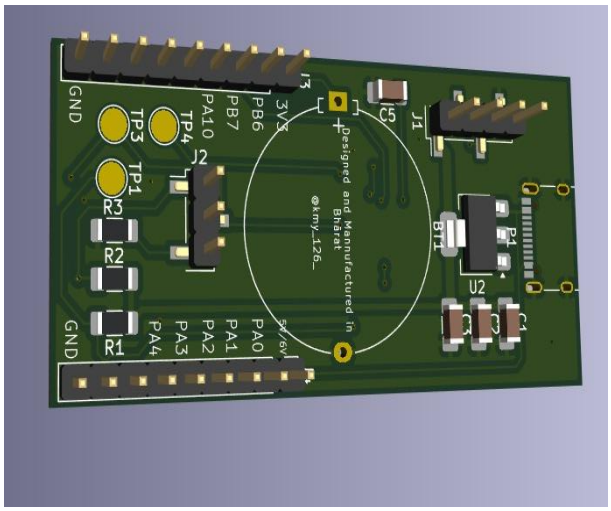

Figure 4.7: Ultra Low Power Module 3D

A USB-C connector as shown in Fig 3.7 is integrated into the design, serving as both a power input and a communication interface, with proper routing for differential data lines and shielding for noise reduction. Together, these elements form a compact yet complete development platform, balancing simplicity with practical features for embedded applications.

V.RESULT AND DISCUSSION.

The Distributed Rail Signalling and Traffic Control System (DRiSTaCS) represents a transformative approach to modern railway operations, addressing the limitations of legacy centralized infrastructures. Traditional systems often rely on fixed block signalling and centralized control centers, which introduce inefficiencies, delays, and vulnerabilities to single-point failures. DRiSTaCS, by contrast, disperses control across multiple autonomous nodes, each capable of processing local sensor data and executing real-time algorithms. This distributed architecture ensures that trains, trackside signals, and control units can communicate seamlessly through encrypted TCP/IP protocols, thereby enhancing both safety and reliability. By integrating fixed and moving block signalling paradigms, the system adapts dynamically to traffic density, maximizing track utilization while maintaining stringent safety standards.

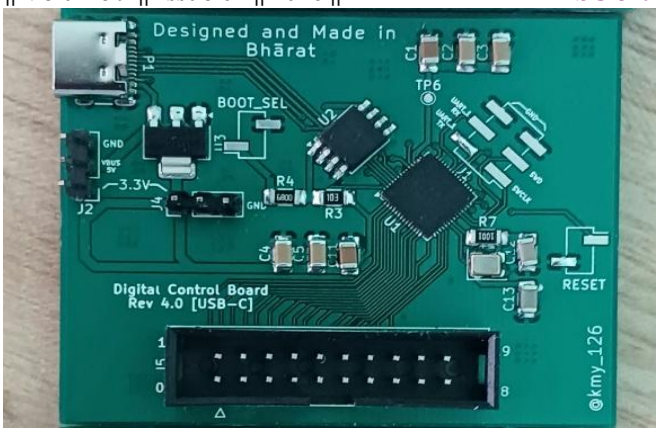

Fig . Digital Control Board

A key strength of DRiSTaCS lies in its robust communication infrastructure. Operating within a closed, secure network, the system restricts access to authenticated devices, minimizing exposure to cyber threats. Wireless communication in the 850 MHz to 2.1 GHz spectrum ensures both long-range penetration and high data throughput, critical for diverse environments ranging from urban stations to rural tunnels. Redundant channels, such as GSM/GPRS modules, provide fallback options to guarantee uninterrupted transmission of emergency signals. This layered communication strategy ensures fault tolerance and resilience, even under heavy operational loads or adverse conditions.

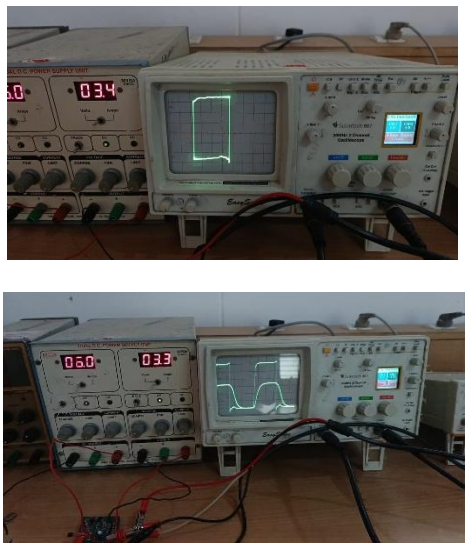

Fig 5.5 1Mhz and 500khz Communication Channel Test

Equally important is the system's integration of multi-modal sensor arrays. Magnetic, optical, and pressure sensors continuously monitor track switches and rail conditions, feeding data into predictive maintenance algorithms. This proactive approach reduces unplanned downtime by scheduling repairs during off-peak hours, shifting rail operations from reactive to preventive models. Furthermore, the self-healing network architecture allows nearby nodes to reroute communications and assume control responsibilities in case of failures, ensuring uninterrupted service. In emergencies, train-to-train communication protocols broadcast encrypted alerts across the network, enabling instantaneous speed adjustments and preventing collisions.

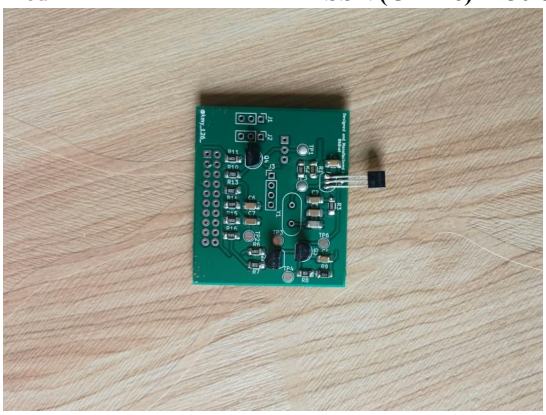

Fig 5.6 12V Communications Module

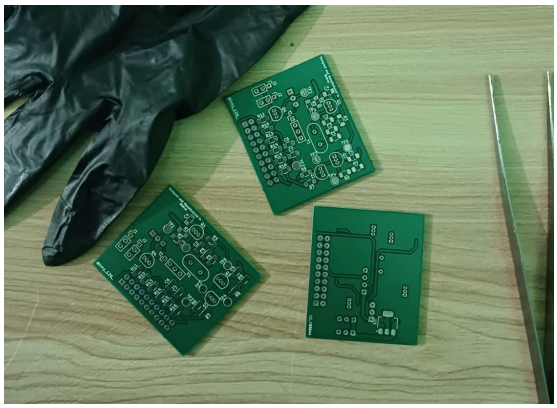

Fig 5.7 Communication Module PCB

The software architecture of DRiSTaCS complements its hardware design. High-performance nodes running Linux handle complex routing and switching, while low-power microcontrollers such as RP2040 execute precise timing tasks in C. Middleware layers in C++ and Node.js orchestrate real-time data parsing and synchronization, while computationally intensive tasks like sensor fusion and route optimization are offloaded to Julia-based nodes. This multi-language ecosystem balances efficiency, scalability, and adaptability, ensuring responsiveness under high data loads.

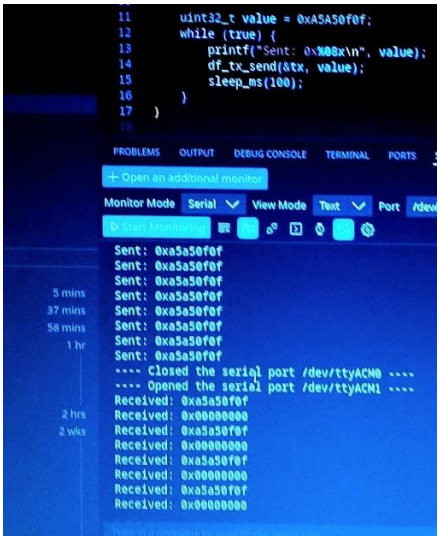

Fig 5.9 Diff_Bus_32 and PIO Reliability Test Result

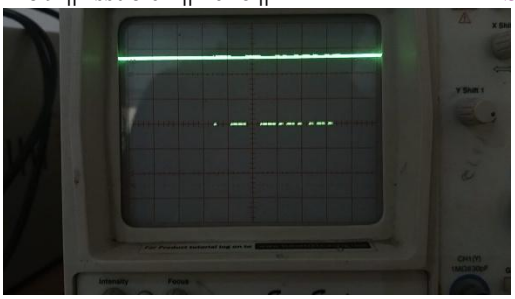

Fig 5.8 Diff_Bus_32 Communication at 560KHz

Ultimately, DRiSTaCS embodies three core objectives: ultra-responsive monitoring, scalable distributed architecture, and minimized human intervention. By automating routine tasks and leveraging advanced encryption, the system reduces human error, lowers operational costs, and enhances safety. In doing so, it paves the way for future-ready rail networks capable of integrating IoT, predictive analytics, and edge computing, ensuring that railway systems remain efficient, reliable, and secure in the decades ahead.

VI.CONCLUSION

The proposed DRiSTaCS framework demonstrates that distributed intelligence can substantially enhance safety, reliability, and scalability in railway operations compared with legacy centralized signalling infrastructures. By decomposing the control plane into autonomous nodes embedded within trains, trackside signals, switches, and maintenance modules, the system eliminates single-point failure modes and shortens decision loops in time-critical scenarios such as emergency braking and route divergence. The hybrid adoption of fixed block and moving block strategies allows dense urban corridors and long-haul segments to be managed under a unified architecture, maximizing line capacity without compromising mandated safety envelopes. A major contribution lies in the tight coupling of sensing, communication, and predictive maintenance. Multi-modal sensor networks continuously validate switch positions and track health, while logged telemetry supports algorithms that anticipate component degradation and schedule downtime-efficient interventions. This approach reduces unscheduled outages and aligns maintenance with operational windows, providing tangible improvements in availability and life-cycle cost. On the communication side, encrypted TCP/IP over licensed and unlicensed bands, backed by GSM/GPRS redundancy, ensures deterministic, low-latency exchanges even under adverse propagation or partial infrastructure failures. The hardware–software co-design further strengthens system robustness. High-performance Linux nodes handle routing and protocol orchestration, whereas RP2040- and STM32-based modules execute tightly bounded real-time tasks with dedicated oscillator and PLL subsystems for timing integrity. Custom ultra-low-power and discrete communication boards demonstrate that reliable signalling can be sustained across diverse power and deployment profiles, enabling incremental rollout on existing lines. Experimental evaluations of digital control boards, differential buses, and communication channels validate the feasibility of the architecture and its readiness for scale-up.Future work can extend DRiSTaCS toward formal safety certification, interoperability with CBTC/ETCS standards, and integration of advanced analytics at the edge for dynamic speed profiling and traffic forecasting. With these enhancements, the system can serve as a foundational platform for smart, autonomous rail corridors in increasingly complex mobility ecosystems.

VII.REFERENCE

[1]. J. Luo, Y. Chen, M. Wu and Y. Yang, "A Survey of Routing Protocols for Underwater Wireless Sensor Networks," in IEEE Communications Surveys & Tutorials, vol. 23, no. 1, pp. 137-160, Firstquarter 2021, doi: 10.1109/COMST.2020.3048190.

[2]. X. Luo et al., "A Lightweight Privacy-Preserving Communication Protocol for Heterogeneous IoT Environment," in IEEE Access, vol. 8, pp. 67192-67204, 2020, doi: 10.1109/ACCESS.2020.2978525.

[3]. Y. Si, N. Korada, R. Ayyanar and Q. Lei, "A High Performance Communication Architecture for a Smart Micro-Grid Testbed Using Customized Edge Intelligent Devices (EIDs) With SPI and Modbus TCP/IP Communication Protocols," in IEEE Open Journal of Power Electronics, vol. 2, pp. 2-17, 2021, doi: 10.1109/OJPEL.2021.3051327.

[4]. S. Babu, A. Rajeev and B. S. Manoj, "A Medium-Term Disruption Tolerant SDN for Wireless TCP/IP Networks," in IEEE Transactions on Network and Service Management, vol. 17, no. 4, pp. 2318-2334, Dec. 2020, doi: 10.1109/TNSM.2020.3023889.

[5]. W. Anani and A. Ouda, "A Secure Lightweight Wireless M-Bus Protocol for IoT: Leveraging the Noise Protocol Framework," *IEEE Canadian Journal of Electrical and Computer Engineering*, vol. 47, no. 4, pp. 175-181, Fall 2024. DOI: 10.1109/ICJECE.2024.3409156]

[6]. H. Li et al., "A Dual Module Parallel Readout System Based on 10 Gb TCP/IP Transmission for HEPS-BPIX Detector," in IEEE Transactions on Nuclear Science, vol. 68, no. 11, pp. 2624-2629, Nov. 2021, doi: 10.1109/TNS.2021.3119028.

[7]. K. Mahmood, S. Shamshad, S. Kumari, M. K. Khan and M. S. Obaidat, "Comment on “Lightweight Secure Message Broadcasting Protocol for Vehicle-to-Vehicle Communication”," in IEEE Systems Journal, vol. 15, no. 1, pp. 1366-1368, March 2021, doi: 10.1109/JSYST.2020.3029895.

[8]. U. H. Hashmi et al., "Effects of IoT Communication Protocols for Precision Agriculture in Outdoor Environments," in IEEE Access, vol. 12, pp. 46410-46421, 2024, doi: 10.1109/ACCESS.2024.3381522.

[9]. Y. Dong, Y. Song and G. Wei, "Efficient Model-Predictive Control for Networked Interval Type-2 T–S Fuzzy System With Stochastic Communication Protocol," in IEEE Transactions on Fuzzy Systems, vol. 29, no. 2,

pp. 286-297, Feb. 2021, doi: 10.1109/TFUZZ.2020.3004192.

[10]. H. Diaz and P. Poór, "Enhancing Industrial Automation: A Practical Study on Communication Protocols and EdMES Software Integration," in IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, vol. 19, pp. 361-370, 2024, doi: 10.1109/RITA.2024.3501218.

[11]. Lavanya Vaishnavi D. A., and A. Kumar C, "Evaluating Supervised Learning Classifier Performance for OFDM Communication in AWGN-Impacted Systems," Results in Engineering, vol. 26, 105178, May 2025, doi: 10.1016/j.rineng.2025.105178.

[12]. S. Bhargavi, R. Verma, A. Kumar C, Lavanya Vaishnavi D. A., J. H. R, and H. S, "Evaluating BER and Throughput in 5G Networks Using Adaptive Modulation and Nakagami Fading," Journal of Information Systems Engineering and Management, vol. 10, no. 3, 2025,

[13]. B. P. Pradeep Kumar, L. Shrinivasan, V. K. R, H. S, Lavanya Vaishnavi D. A., and J. H. R, "AI-Powered Mask Uncovering in Complex Occluded Environments," Journal of Information Systems Engineering and Management, vol. 10, no. 3, pp. 958–970, 2025. doi: Not provided.

[14]. B. P. Pradeep Kumar, G. R. Poornima, S. H. S., Anil Kumar C, and Lavanya Vaishnavi D. A., "Deep Learning-Based Approach for Early Detection of Osteoporosis Using X-ray Imaging," Computer Fraud and Security, vol. 2024, no. 12, pp. 384–398, 2024.