TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AAA Aaa Aaa Aaa

BB Bbb Bbb

CCC Ccc Ccc Ccc

GENERAL INTRODUCTION

This project focuses on exploring the intricacies of Controller Area Network (CAN) communication, critical for ensuring robust data exchange in automotive and industrial applications. The study comprises both theoretical analysis and practical implementation to comprehensively understand CAN system dynamics.

The theoretical segment begins with a detailed review of current CAN communication methodologies, emphasizing timing analysis and critical parameters affecting signal integrity and data reliability. Using LTspice, the CAN Bus is modeled as a transmission line to simulate various operational conditions and evaluate the impact of noise, impedance mismatches, signal reflections, and termination issues on communication performance.

Practical implementation involves two main phases : demonstration and measurement of CAN communication issues. The demonstration phase focuses on setting up and executing CAN communication between TriCore boards. This includes identifying necessary hardware components and measurement tools to facilitate data transmission scenarios, such as reading from a temperature sensor and displaying values on an LCD via CAN.

The subsequent phase centers on measuring and analyzing common CAN communication challenges using an oscilloscope. Key issues evaluated include improper adaptation, stub connections, noise interference, unmatched line lengths, electromagnetic interference (EMI), and impedance mismatches. The project systematically varies CAN transmission rates to assess their impact on critical performance metrics like message latency, throughput, error rates, and jitter.

The overarching goal of this project is to advance understanding of CAN communication complexities and develop strategies to optimize CAN Bus performance and reliability in practical environments.

Chapitre
1

Theoretical Study

Sommaire

1.1 INTRODUCTION

The first section of this report focuses on the temporal characteristics and operational aspects of the CAN (Controller Area Network) bus. As a fundamental communication protocol in automotive and industrial applications, understanding the temporal dynamics of the CAN bus is crucial for ensuring reliable data transmission. This section explores key elements such as frame types, propagation delays, and the mechanisms involved in error detection and correction. By comprehensively analyzing these aspects, this section aims to provide a foundational understanding of how temporal factors influence the performance and reliability of CAN bus communication in real-world environments.

1.2 Analysis of the current state

The figure 1.1 shows that the absence of a Controller Area Network (CAN) bus in modern vehicles can lead to significant drawbacks. Without a CAN bus, vehicles rely on individual wiring connections between electronic components, increasing wiring complexity and the risk of faults. This can also limit real-time data exchange between systems, impacting the implementation of advanced features like diagnostics and engine control. Additionally, the absence of a CAN bus can make maintenance more challenging, as technicians may struggle to diagnose and repair wiring issues efficiently. Overall, the lack of a CAN bus can reduce reliability, limit flexibility in adding new features, and increase maintenance costs in vehicle electrical systems.

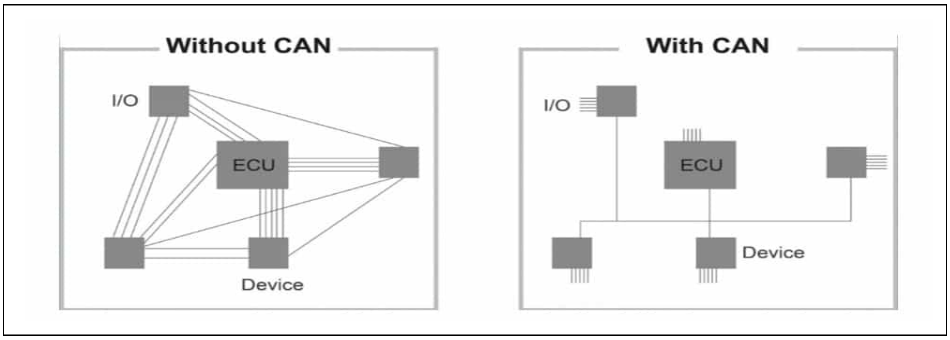

FIGURE 1.1 – Network architecture before and after the CAN

1.3 Time Analysis of the CAN Bus

1.3.1 Types of Frames in the CAN Bus

The CAN bus uses several types of frames, each with a specific role and impact on the timing of communications :

• Data Frame : Used for the transmission of data between network nodes, as shown in the figure 1.2.

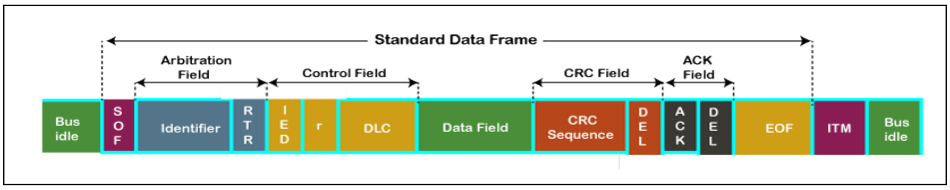

FIGURE 1.2 – Data Frame

• Remote Frame : Used to request data from another node, as shown in the figure 1.3.

.

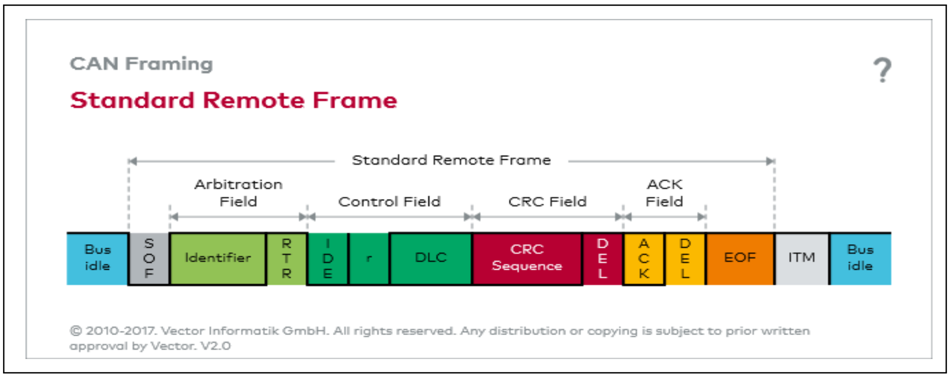

FIGURE 1.3 – Remaote Frame

• Overload Frame : Indicates a delay needed to process an overload, as shown in the figure 1.4..

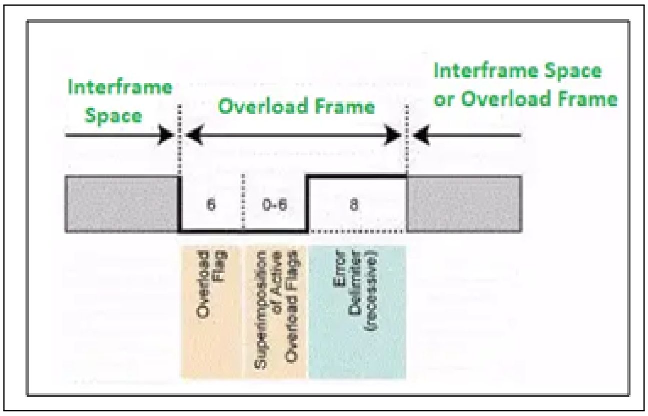

FIGURE 1.4 – Overload Frame

• Error Frame : Sent in case of error detection to notify all nodes of the presence of an error, as shown in the figure 1.5.

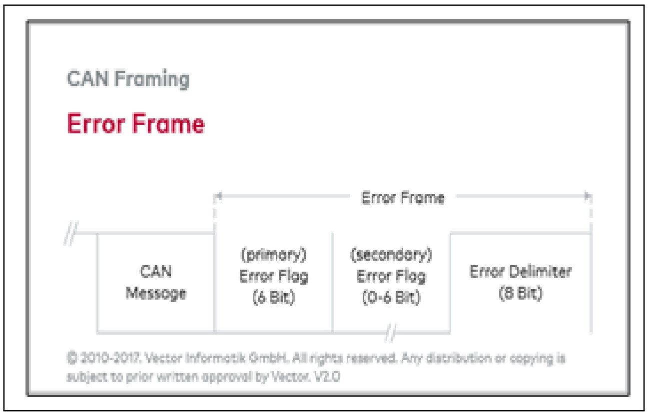

FIGURE 1.5 – Error Frame

• Inter-Frame Space (IFS) : used to delay the next packet data transmission, as shown in the figure 1.6.

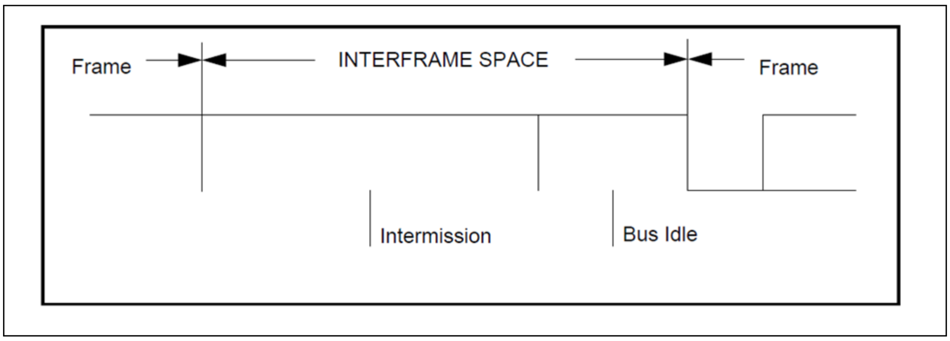

FIGURE 1.6 – Inter-Frame Space

1.3.2 Multi-Master Architecture

The CAN bus operates on a multi-master architecture, where each node can initiate communication. This capability requires an effective arbitration mechanism to manage conflicts when multiple nodes attempt to transmit simultaneously, as shown in the figure 1.7.

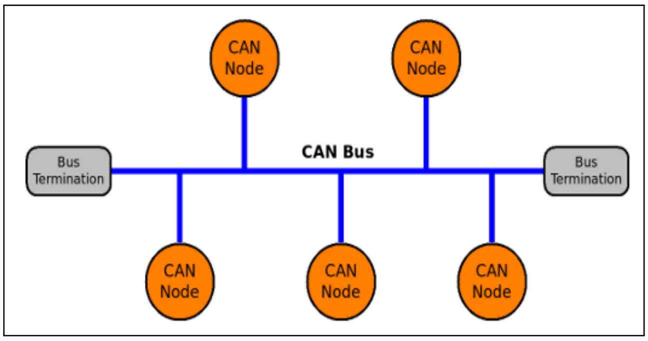

FIGURE 1.7 – Multi-Master Architecture

1.3.3 Transmission Suspension

.

Transmission suspension occurs when errors are detected or when the bus is occupied by a higher-priority frame. Nodes must wait for an inter-frame period before retrying transmission.

1.3.4 MSB and LSB Bits

• MSB (Most Significant Bit) : The leftmost bit, representing the highest value in a binary sequence. • LSB (Least Significant Bit) : The rightmost bit, representing the lowest value in a binary sequence.

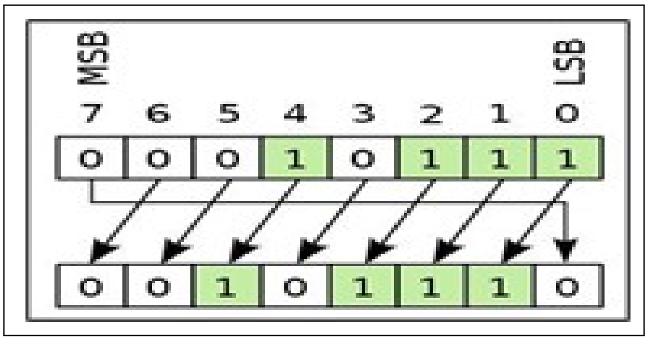

FIGURE 1.8 – LSB and MSB

1.3.5 Propagation Delays and Bus Length

.

The length of the CAN bus and the propagation delay of signals are critical factors that influence the overall timing of the network : • Propagation Delay : The time it takes for a signal to travel from one end of the bus to the other. This delay is influenced by the bus length and the transmission medium. • Bus Length : There is an inverse relationship between the maximum bus length and the transmission rate. As the bus length increases, the transmission rate must be reduced to ensure reliable communication, as shown in Figure 1.9.

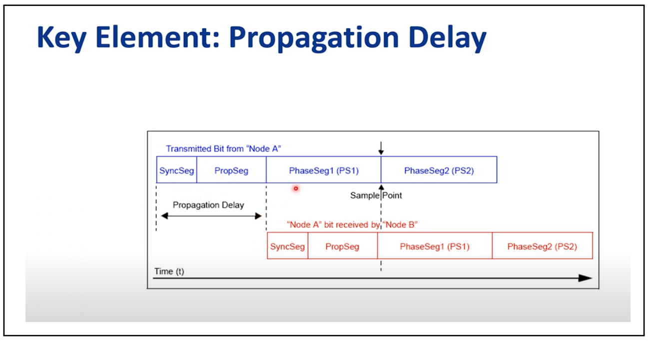

FIGURE 1.9 – Propagation Delay

1.3.6 Coding Techniques and Stuffing Bits

• NRZ (Non-Return to Zero) Coding : Used in the physical layer to transmit data without returning to zero between bits, reducing signal transitions and energy consumption, as shown in Figure 1.10.

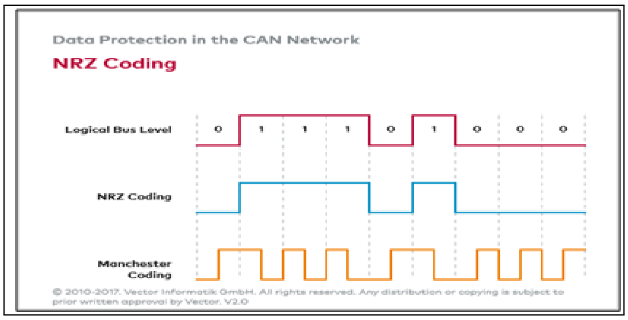

FIGURE 1.10 – NRZ (Non-Return to Zero) Coding and the other types of coding

• Stuffing Bits : Added to avoid long sequences of identical bits, which can complicate synchronization and error detection, as shown in figure 1.11.

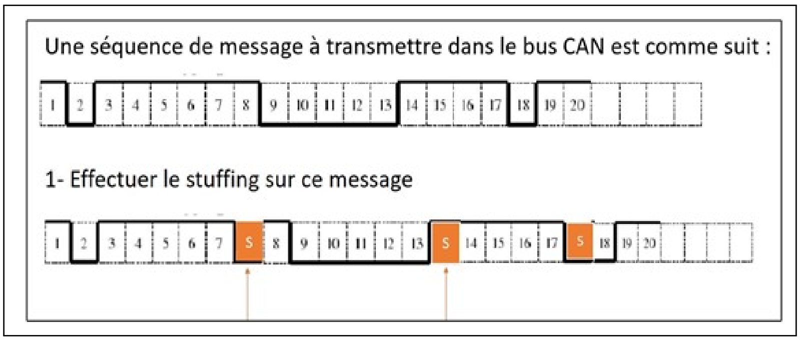

FIGURE 1.11 – Stuffing bit

1.3.7 End-of-Line Resistors

To prevent signal reflection, termination resistors (often mistakenly referred to as pull-up resistors) are placed at the ends of the CAN bus. These resistors match the characteristic impedance of the bus and dissipate signals to prevent interference.

1.3.8 Error Management

• Error Flag : Used to indicate the detection of an error, leading to the transmission of an error frame.

• Error Delimiter : A field of bits following the error, allowing separation of error frames from data frames.

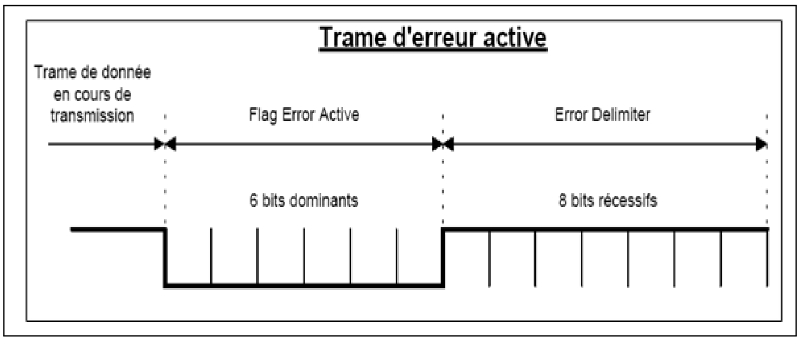

FIGURE 1.12 – Error Trame

1.3.9 CAN Functional Units

• **BitStream Processor** : Manages the transmission and reception of bits on the bus.

• **Error Handling Unit** : Detects and signals errors, activating the error flag if necessary.

• **Bit Timing Unit** : Ensures bits are transmitted and received at correct intervals, synchronizing network nodes.

• **Interrupt Control Unit** : Manages interrupts, allowing nodes to respond to priority events.

• **Frame Counter** : Keeps track of the number of frames transmitted and received, helping in monitoring and diagnostics of the communication process.

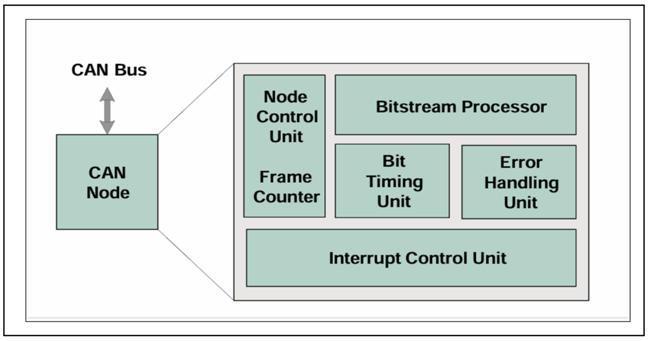

FIGURE 1.13 – Node composition

1.3.10 CSMA Techniques

.

Carrier Sense Multiple Access (CSMA) protocols manage access to shared communication media. This section covers the main types of CSMA protocols : Persistent CSMA, Non-Persistent CSMA, P-Persistent CSMA, CSMA/CD (Collision Detection), CSMA/CD + AMP (Adaptive Minimum Penalty), and CSMA/CA (Collision Avoidance), detailing their operations and applications.

Protocol	Operation	Advantages	Disadvantages
Persistent CSMA	Transmits immediately if idle, keeps listening if busy	Efficient under low load	High collision probability under high load
Non-Persistent CSMA	Waits random period if busy, transmits immediately if idle	Reduces collision probability	Higher delays due to random wait
P-Persistent CSMA	Transmits with probability PPP if idle, repeats if busy	Balances immediate and random wait	Still possible collisions under high load
CSMA/CD	Aborts on collision, waits random backoff, and retries	Reduces collision time wastage	Ineffective for wireless networks
CSMA/CD + AMP	Adaptive backoff period based on network conditions	Better performance under dynamic conditions	Increased complexity
CSMA/CA	Uses interframe space and RTS/CTS for collision avoidance	Effective for wireless networks	Adds overhead with control messages

TABLE 1.1 – Protocols Comparison

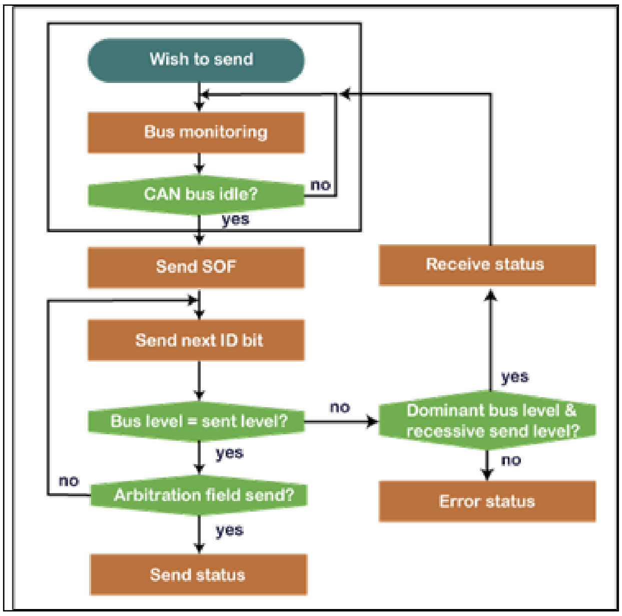

FIGURE 1.14 – The Mechanism of sending frames

=> The bus in these networks can be in one of two states : Carrier Busy, indicating that a transmission is ongoing, or Carrier Idle, indicating no transmission is currently occurring on the bus. However, despite this listening mechanism, the possibility of collision still exists due to propagation delay. This delay means that a node may sense the bus as idle, but in reality, this is because the first bit sent by another node has not yet reached it.

1.3.11 Maximum Frame Size

The maximum size of a data frame on the CAN bus is 64 bits, including the fields for identifier, control, data, CRC, ACK, and end of frame.

1.3.12 Communication Delay Analysis

To analyze communication delays on the CAN bus, we must consider transmission times, arbitration delays, and interframe periods :

• Transmission Time : Calculated based on frame size and transmission rate.

Débit	Longueur	Longueur d'un bit
1 Mbit/s	30 m	1 µs
800 kbit/s	50 m	1,25 µs
500 kbit/s	100 m	2 µs
250 kbit/s	250 m	4 µs
125 kbit/s	500 m	8 µs
62,5 kbit/s	1000 m	16 µs
20 kbit/s	2500 m	50 µs
10 kbit/s	5000 m	100 µs

FIGURE 1.15 – Length of a Bit According to Debit

• Arbitration Delays : Depend on frame priority and the bus's arbitration mechanism.

.

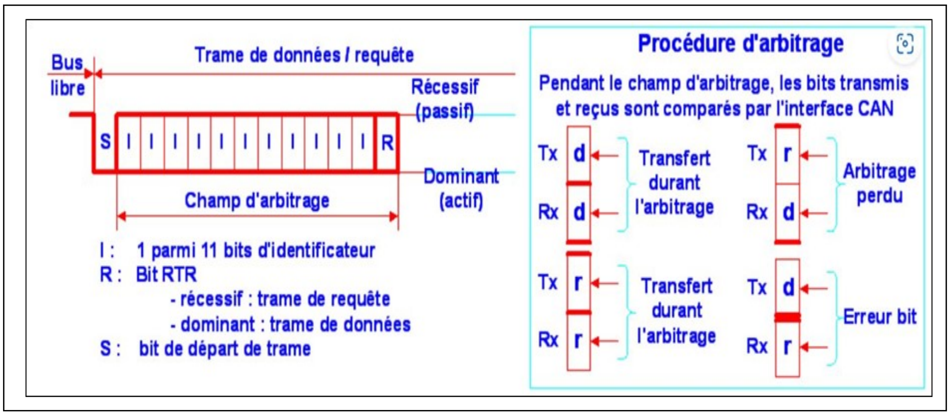

FIGURE 1.16 – CAN Bus Arbitration Process

• Interframe Periods : Add a fixed delay between frames to ensure orderly communication and.node synchronization.

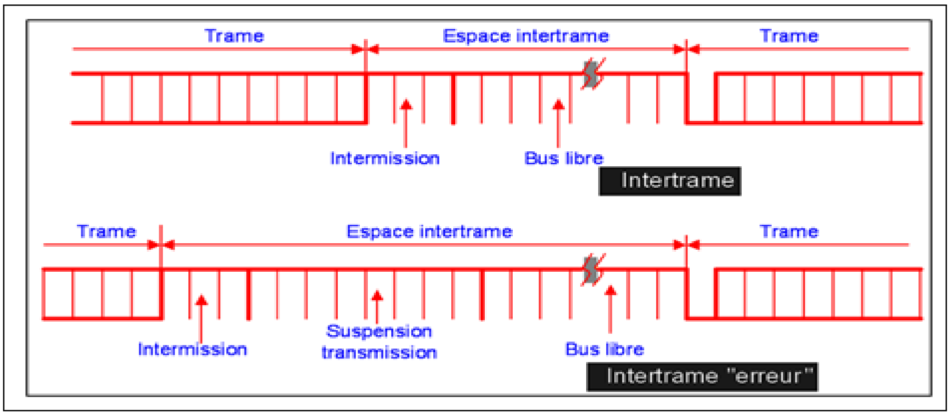

FIGURE 1.17 – Diagram of a CAN Bus Interchange

1.3.13 Arbitration Process in CAN

In a Controller Area Network (CAN), multiple nodes can try to transmit data simultaneously. To manage these simultaneous transmission attempts, CAN uses an arbitration process based on the CAN protocol. This process ensures that only one node can transmit at a time, preventing data collisions and ensuring efficient communication.

How Arbitration Works :

1. **Message Priority** : Each message in a CAN network has a unique identifier, which also determines its priority. Lower identifier values represent higher priority messages

2. **Bitwise Arbitration** : When multiple nodes start transmitting simultaneously, they send their identifiers bit by bit. The CAN bus uses a dominant-recessive bit model, where a dominant bit (0) can overwrite a recessive bit (1).

3. **Dominant Bits Win** : If a node sends a recessive bit but detects a dominant bit on the bus, it realizes that a higher priority message is being transmitted. It will then stop its transmission and wait until the bus is free again.

4. **Continued Transmission** : The node with the highest priority (lowest identifier) continues to transmit without interruption. Other nodes with lower priority identifiers will wait and attempt to transmit once the bus is available.

Example Scenario :

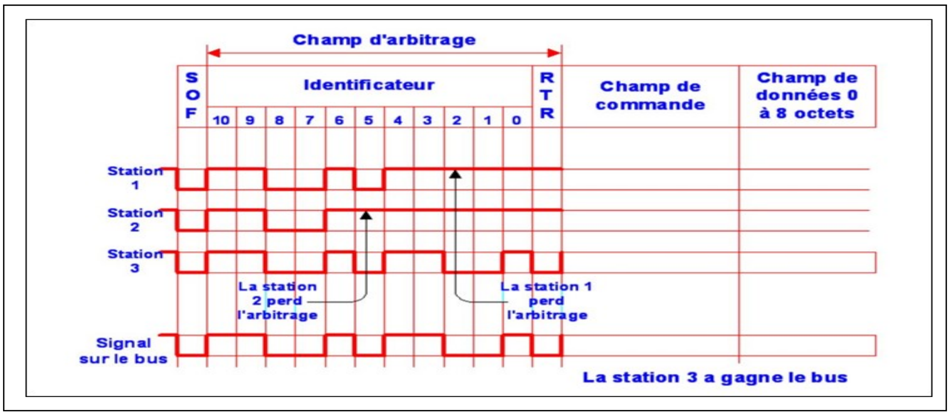

FIGURE 1.18 – CAN Bus Frame Transmission

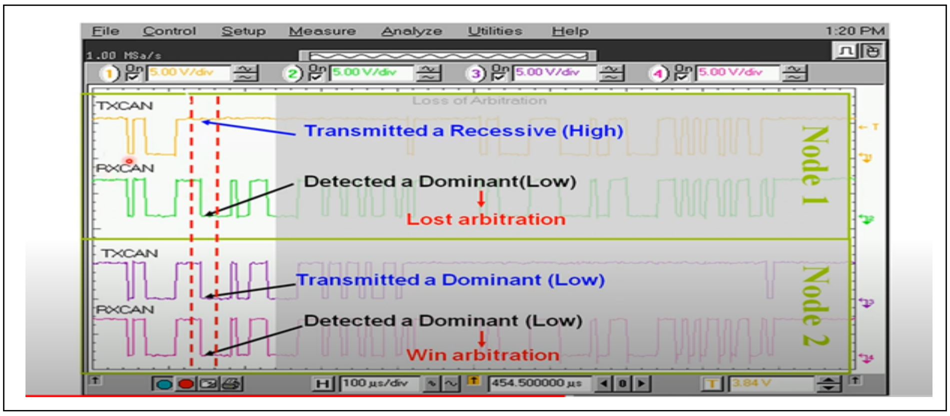

FIGURE 1.19 – Failure to Win Arbitration on the CAN Bus

1.3.14 CAN Frame Fields

The figure 1.20. shows the different compositions of the Extended CAN frame :

• SOF (Start of Frame) : Indicates the beginning of the frame.

• Arbitration Field : Contains the frame identifier and the RTR (Remote Transmission Request) control bit.

• Control Field : Indicates the length of the data.

• Data Field : Contains the data to be transmitted (up to 8 bytes).

• CRC Field : Used for error detection.

• ACK Field : Indicates if the frame was correctly received.

• EOF (End of Frame) : Indicates the end of the frame.

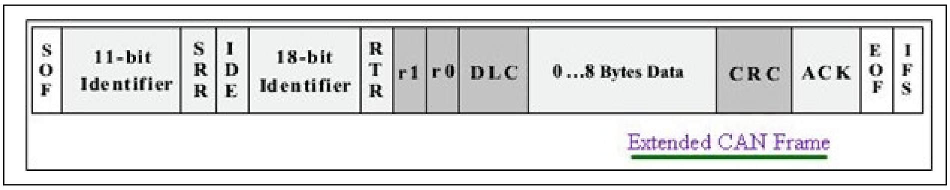

FIGURE 1.20 – Extended CAN Frame Format

1.3.15 Functioning of CANL and CANH

The figure 1.21. shows the difference in potential between CANL and CANH :

• CANL (CAN Low) : Low voltage signal.

• CANH (CAN High) : High voltage signal.

In differential mode, both signals are used to improve noise immunity and enable reliable communication.

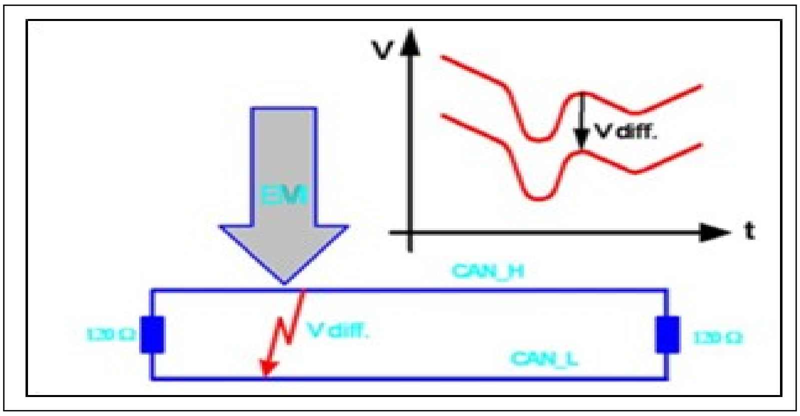

FIGURE 1.21 – The difference in potential between CANH (CAN High) and CANL (CAN Low)

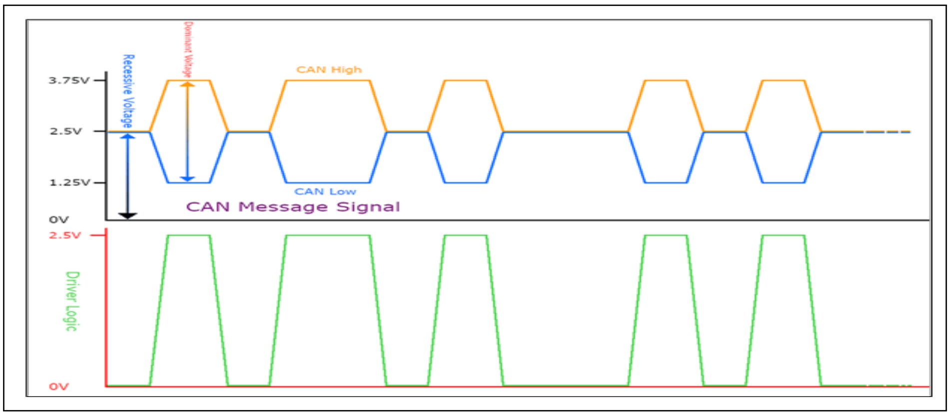

FIGURE 1.22 – CAN Bus Arbitration Process

1.3.16 CAN Transmission Types

There are three types of transmission in CAN networks as presented in Table 1.2 : CAN Low Speed, CAN High Speed, and CAN FD (Flexible data rate). The following table provides a comparison of these types :

Parameter	CAN Low Speed	CAN High Speed	CAN FD
Débit	125 kb/s	125 kb/s to 1 Mb/s	500 kb/s to 8 Mb/s
Nombre de nœuds	2 to 20	2 to 30	2 to 30
Courant de sortie	> 1mA on 2.2 k	25 to 50 mA on 60	25 to 50 mA on 60
Niveau dominant	CAN H = 4V, CAN L = 1V	CAN H = 3.5V, CAN L = 1.5V	CAN H = 3.5V, CAN L = 1.5V
Niveau récessif	CAN H = 1.75V, CAN L = 3.25V	CAN H = CAN L = 2.5V	CAN H = 2.5V, CAN L = 1.5V
Caractéristiques du câble	30 pF for CAN H and CAN L	2 * 120 for CAN H and CAN L	2 * 120 for CAN H and CAN L
Tensions d'alimentation	5V	5V	5V

TABLE 1.2 – CAN type Comparison

1.3.17 FIFO Protocol

The CAN bus uses a FIFO (First In, First Out) protocol to manage frame queues. This means that frames are transmitted in the order they arrive, ensuring fair data processing.(Figure 1.23)

FIGURE 1.23 – The principle of frame transmission in CAN

1.3.18 CAN Node Configurations

.

In a CAN (Controller Area Network) system, each node typically comprises the following components (as shown in Figure 1.24) :

- A CAN Transceiver
- A CAN Controller
- An Application Controller (MCU)

The nodes are configured in various ways to suit different application requirements.

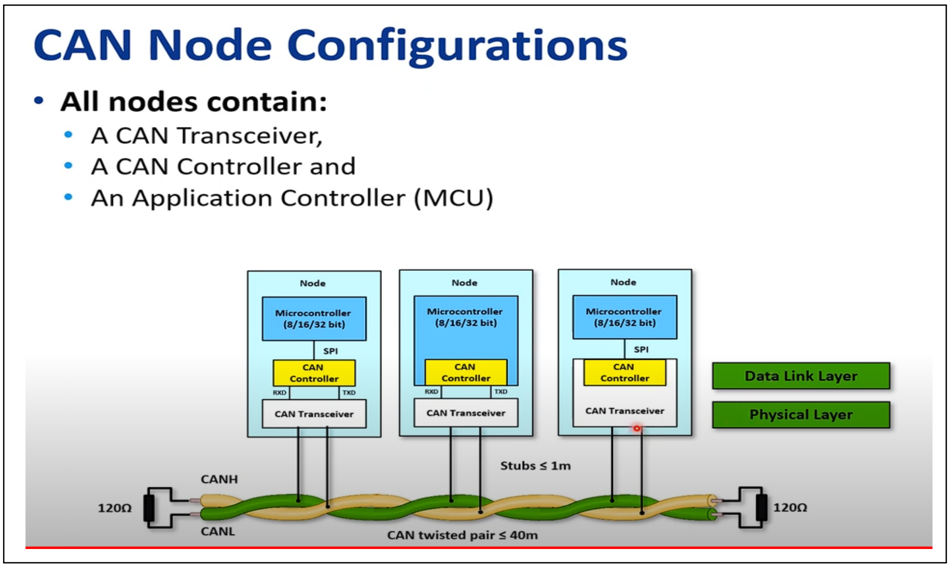

FIGURE 1.24 – CAN Node Block Diagram

1.3.19 Advantages and disadvantages of CAN bus

In this following Table 1.3 are some pros and cons of CAN bus over other bus types.

Advantages	Disadvantages
Low use of wiring due to its distributed control	Limited length
Can be applied to many electrical environments without any issues.	Network must be wired in topology that limits stubs as much as possible.
Multi master and multicast features can be applied	Undesirable interaction between nodes
High speed data rate	Limited number of nodes (up to 64 nodes)
Low cost and light in weight and robustness	High cost for software development and maintenance
Supports auto retransmission for attribution lost messages	Possibility of signal integrity issues
Built in error detection capabilities. (ack error, form error stuff error etc.)	

TABLE 1.3 – Advantages and disadvantages of CAN bus

1.4 JTAG and SWD Interfaces

1.4.1 JTAG (Joint Test Action Group) Interface

The JTAG interface is used for programming and debugging integrated circuits. It allows access to the device's internal registers and memory, facilitating testing and debugging :

• 14 Pins : The JTAG interface has 14 pins for connection and control.

• TAP (Test Access Port) : The main port used for JTAG communication.

• Scan Chain : Allows sequential testing of the circuit's pins.

• JTAG Instructions : Include standardized commands for executing tests and controlling registers.

• Adaptable to Various Architectures : Can be used with a variety of microcontrollers and processors.

• 4 Required Pins (TMS, TCK, TDI, TDO) : For basic communication, only four pins are essential.

• Boundary Scan : Used to test connections and integrated circuits by verifying signal path continuity.

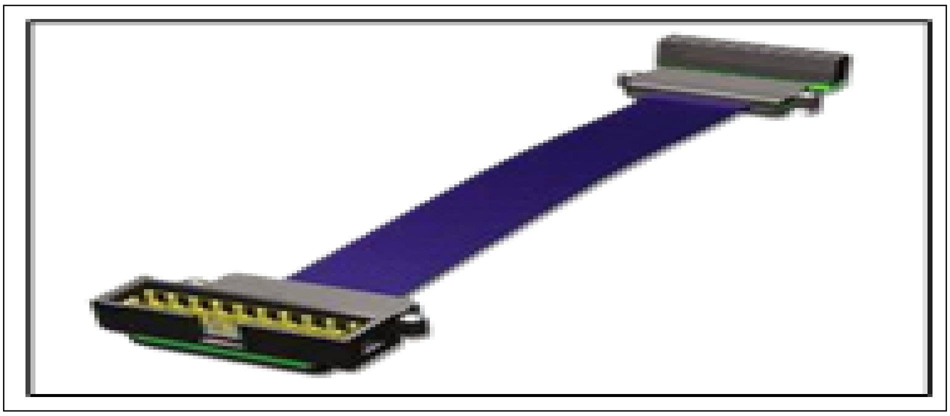

FIGURE 1.25 – Joint Test Action Group

.

1.4.2 SWD (Serial Wire Debug) Interface

SWD is a simpler and faster alternative to JTAG for debugging and programming microcontrollers :
• 2 Pins (SWDIO, SWDCLK) : Uses only two lines, one for the clock (SWCLK) and one for data (SWDIO).
• Bidirectional Wire : The SWDIO line is bidirectional, allowing two-way communication.
• Star Topologies : SWD can be used in star configurations to connect multiple debugging devices.

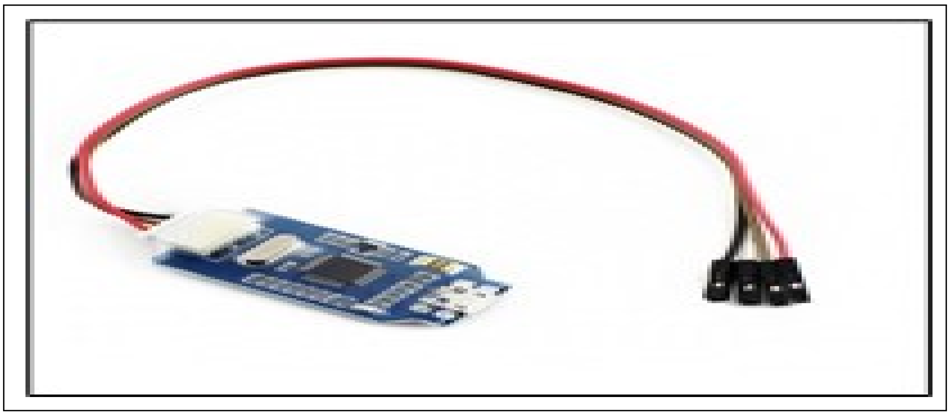

FIGURE 1.26 – Serial Wire Debug

1.5 Applications for the CAN

• **Automotive** : CAN is extensively used in the automotive industry for communication between various control units, such as the engine control unit (ECU), transmission control unit (TCU), airbag control unit, and infotainment system. It enables real-time data exchange, diagnostics, and control, improving vehicle performance, safety, and efficiency.(Figure 1.25)

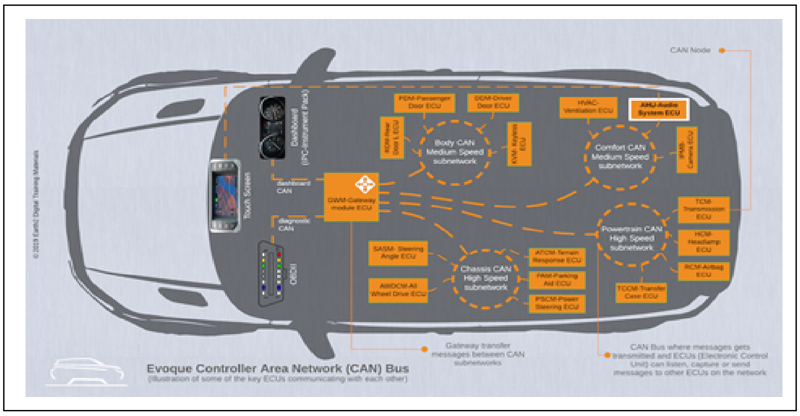

FIGURE 1.27 – CAN in the automotive industry

• **Industrial Automation** : CAN is widely used in industrial automation for communication between programmable logic controllers (PLCs), sensors, actuators, and other devices. It facilitates efficient and reliable data exchange, enabling automation and control systems to operate seamlessly.(Figure 1.26)

FIGURE 1.28 – CAN in the industrial automation

• **Medical Devices** : CAN is employed in medical devices for communication between different components, such as sensors, monitors, and control units. It ensures accurate and timely data transmission, crucial for patient monitoring and treatment.(Figure 1.27)

FIGURE 1.29 – CAN in the Medical field

• **Home Automation** : CAN is increasingly being used in home automation systems for communication between different smart devices, such as thermostats, lights, and security systems. It enables seamless integration and control of various home automation devices.(Figure 1.28)

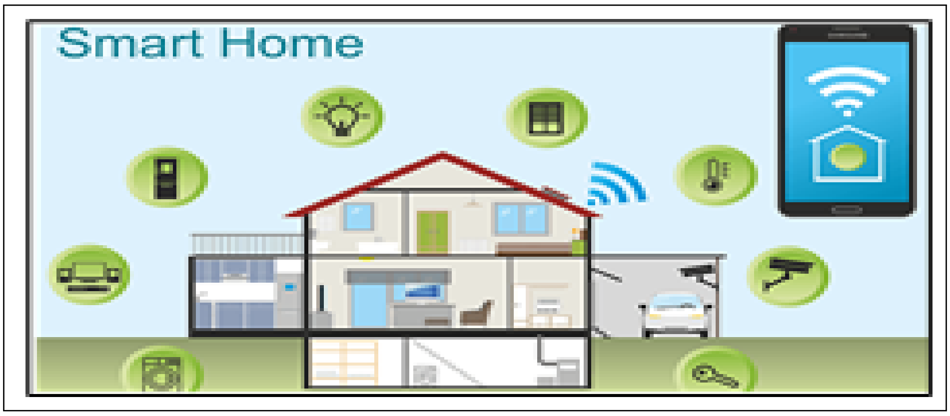

FIGURE 1.30 – CAN in the Home automation

1.6 Modeling and Analysis of CAN Bus Transmission Line in LTspice

1.6.1 Initial Case :

In this initial setup (Figure 1.31), we design a minimalist circuit schema to visualize the signals of the CAN bus line transmission.

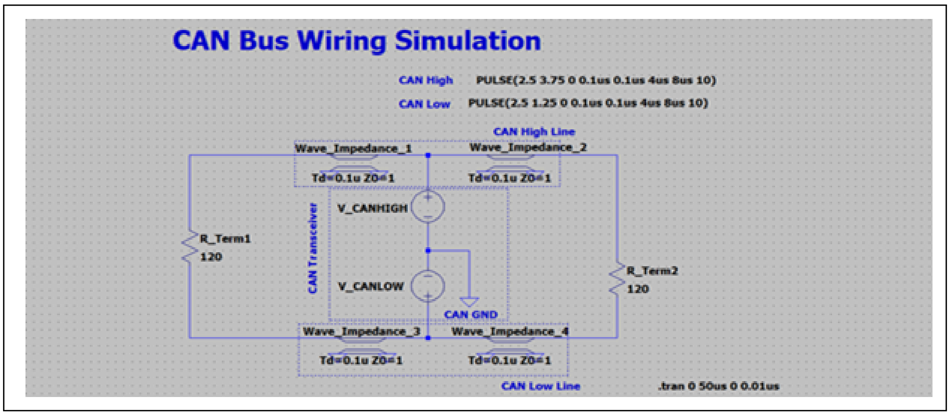

FIGURE 1.31 – The ideal Case

This transmission line setup,in figure 1.31, includes the following components :

• **CAN Node Transceiver** : The transceiver is responsible for sending and receiving the CAN bus signals, converting the data from the CAN controller to differential signals on the bus and vice versa.

• **Two Opposite Direction Generators** : These generators simulate the data transmission in both directions on the CAN bus. They provide the differential signals required for CAN communication.

• **CAN Lines (HIGH LOW)** : The CAN bus consists of two lines, CANH (High) and CANL (Low), which carry the differential signals. These lines are crucial for ensuring noise immunity and reliable data transmission.

• **Two 120-Ohm Resistors** : These resistors are used to terminate the CAN bus. Proper termination with 120-ohm resistors at both ends of the bus is essential to prevent signal reflections and ensure signal integrity.

=> This minimalist circuit provides a clear and straightforward representation of the basic CAN bus line transmission, allowing us to visualize and analyze the behavior of the signals under ideal conditions.

Components of CAN Bus Architecture :

1. Nodes : Devices or components connected to the CAN Bus network, including sensors, controllers, and displays.

2. Messages : Transmitted information in the form of messages. Each message has an identifier (ID) determining its priority and content.

3. Termination : Proper termination with 120-ohm resistors at both ends of the network prevents signal reflections and maintains data integrity.

In this case the signals are visualized like this :

• The potential difference between VCANHIGH and VCANLOW in the perfect case :

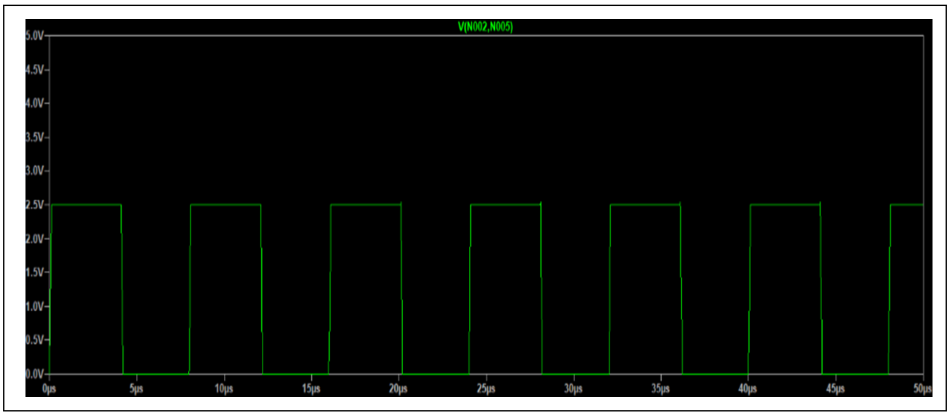

FIGURE 1.32 – The real case

In an ideal scenario, figure 1.32, the potential difference between CANL (Controller Area Network Low) and CANH (Controller Area Network High) is stable and follows the expected waveform precisely. This is essential for ensuring accurate data transmission and communication within the CAN bus system.

• **CANL(Green) CANH(Blue) in the real case without disturbance :**

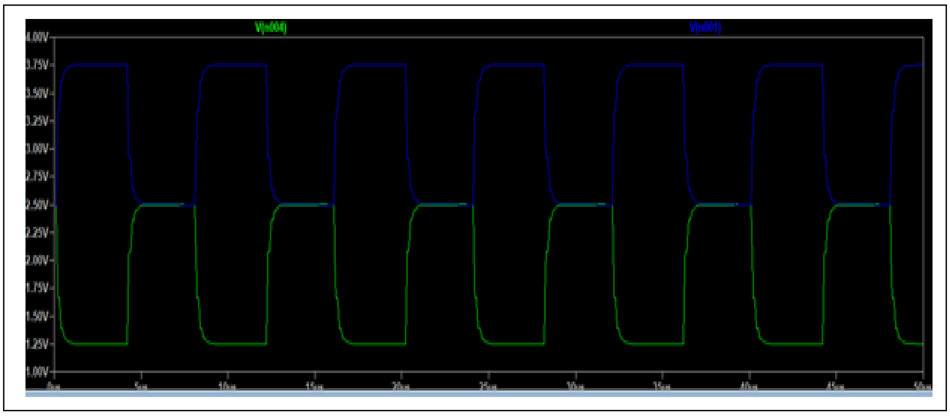

FIGURE 1.33 – The principle of frame transmission in CAN

In a real-world scenario, figure 1.33, even without any external disturbances, the potential difference between CANL and CANH might exhibit slight variations due to inherent imperfections in the components and the environment. However, these variations should be minimal and within the acceptable range defined by the CAN protocol standards.

- **Comparison to the real case :**

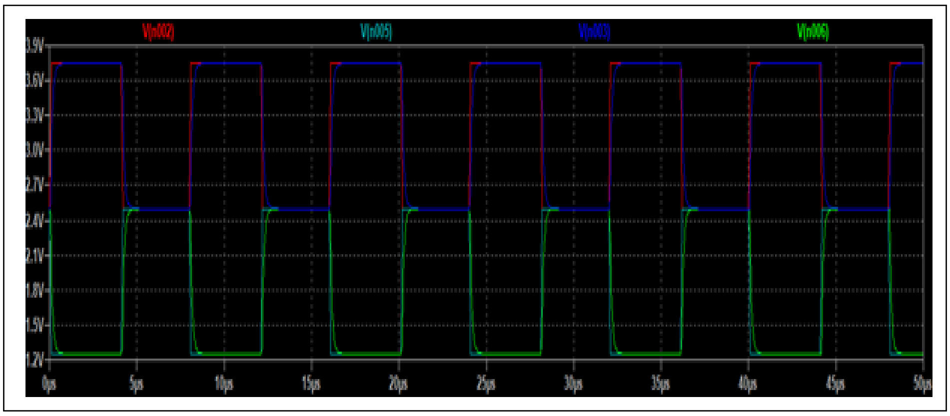

FIGURE 1.34 – The comparison the ideal and real case

Comparing the ideal and real cases highlights the deviations caused by practical limitations. Figure 1.34 shows that These comparisons help in understanding how close the real-world implementation is to the theoretical model and identify any areas that might need improvement to enhance the reliability and performance of the CAN bus system.

1.6.2 Behavior study :

In this simulation, Figure 1.35, we use capacitance and inductance to model disturbances on a transmission line. Starting with low values, we gradually increase them to demonstrate how signal quality deteriorates. As capacitance and inductance rise, the transmitted signal loses its original shape and shows increasing impurities, indicating significant disturbance.

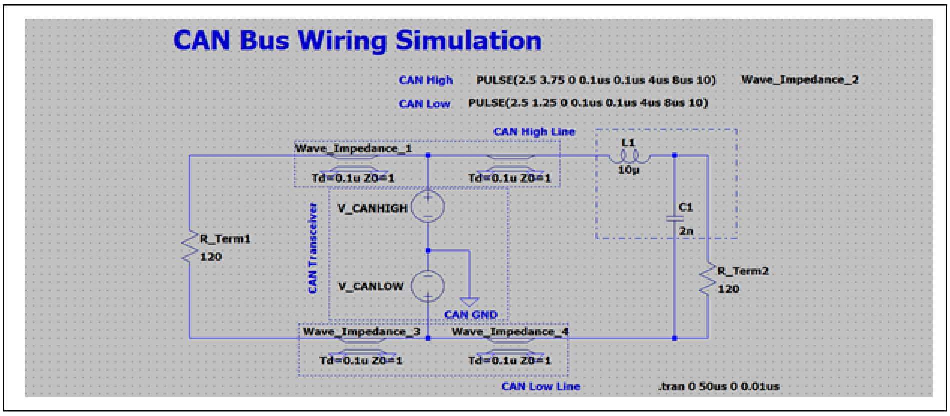

FIGURE 1.35 – A schema of the CAN Bus Line Transmission

These values represent the line's distributed parameters, affecting signal integrity through impedance changes and noise induction. Observing this degradation helps understand the impact of disturbances on transmission lines, emphasizing the importance of mitigating measures such as proper shielding, grounding, and design optimization to maintain reliable communication in CAN Bus and other network systems.

Methodology :

The CAN Bus model is created in LTspice, simulating various scenarios to observe their effects on signal integrity. The scenarios include :

1. Impedance Mismatch

2.Signal Reflections

3.Termination Issues

4. Noise Effects

Simulation Setup :

• Components :

o Transmission Lines : Modeled using LTspice transmission line elements (T1, T2, T3, T4).

o Noise Sources : Voltage sources with added AC components to simulate noise.

o Termination Resistors : Resistors at the end of the transmission lines to terminate the CAN Bus

• Simulation Parameters :

o Pulse Generators : Used to generate the CAN High (CANH) and CAN Low (CANL) signals.

o Impedance Values : Different impedance values are set for the transmission lines to simulate mismatches.

o Noise Levels : Noise sources are adjusted to simulate different levels of external interference.

1.6.2.1 Simulation Results without Noise effect :

A. Impedance Mismatch

• Simulation Setup :

T1 Td=0.1u Z0=100

T2 Td=0.1u Z0=140

T3 Td=0.1u Z0=120

T4 Td=0.1u Z0=120

• Signal Visualization :

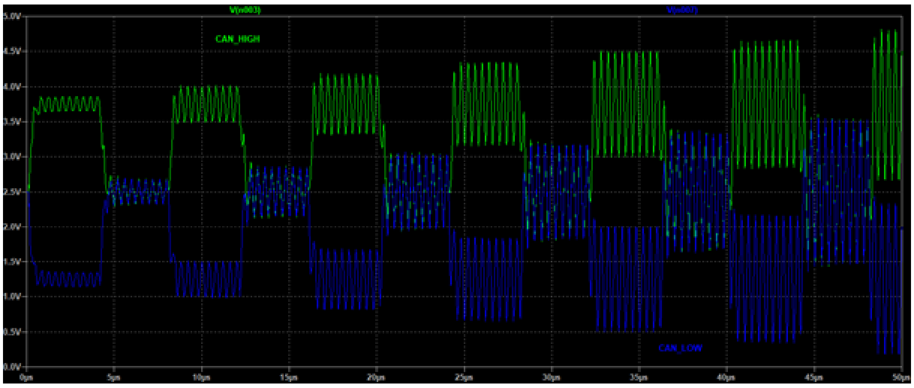

FIGURE 1.36 – The visualization of the effects of the Impedance Mismatch

• FFT Visualization :

CAN LOW FFT visualization :

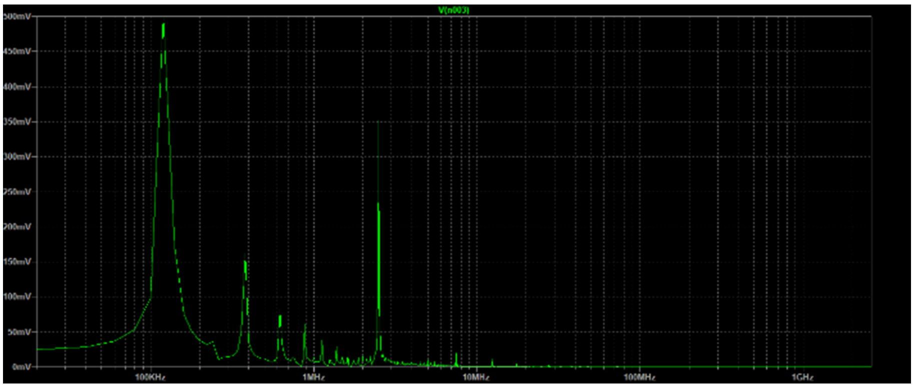

FIGURE 1.37 – The CAN LOW FFT visualization

CAN HIGH FFT visualization :

.

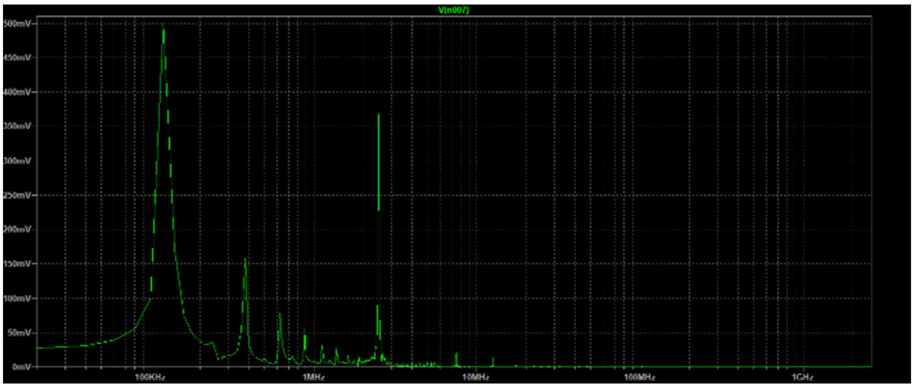

FIGURE 1.38 – The CAN HIGH FFT visualization

• **Waveforms :** o CANH and CANL signals are observed for distortions and reflections.

o FFT Analysis : Fast Fourier Transform (FFT) is applied to analyze the frequency components of the signal.

• **Observation :** The impedance mismatch causes reflections in the signal, leading to distortions and potential communication errors. The FFT analysis shows increased frequency components corresponding to the reflections.

• **Details :**

- Reflection Coefficient Calculation : Based on the impedance mismatch, the reflection coefficient is calculated to quantify the extent of signal reflection.

- Signal Integrity Metrics : Metrics such as signal-to-noise ratio (SNR) and eye diagram analysis are used to evaluate the signal integrity.

1.6.2.2 Signal Reflections

Simulation Setup :

First case : Tstub, T*d=0.05u, Z0=100(resistance - mismatch)

Second case : Tstub T*d=1u, Z0=100

Visualization

1st Case :

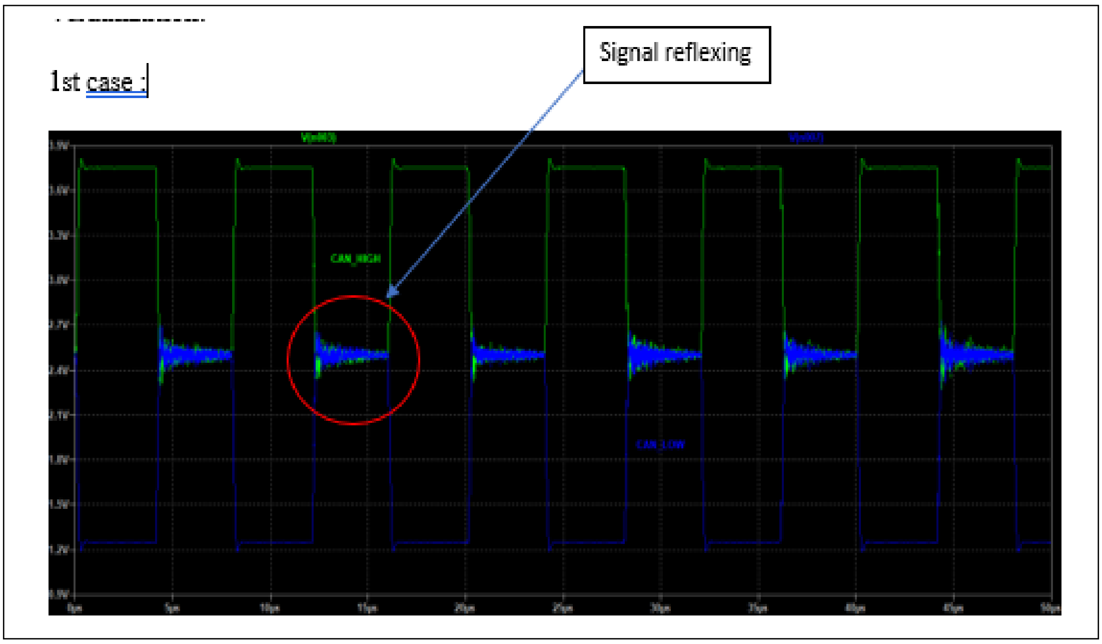

FIGURE 1.39 – 1st case signal reflection visualization

.

2nd case :

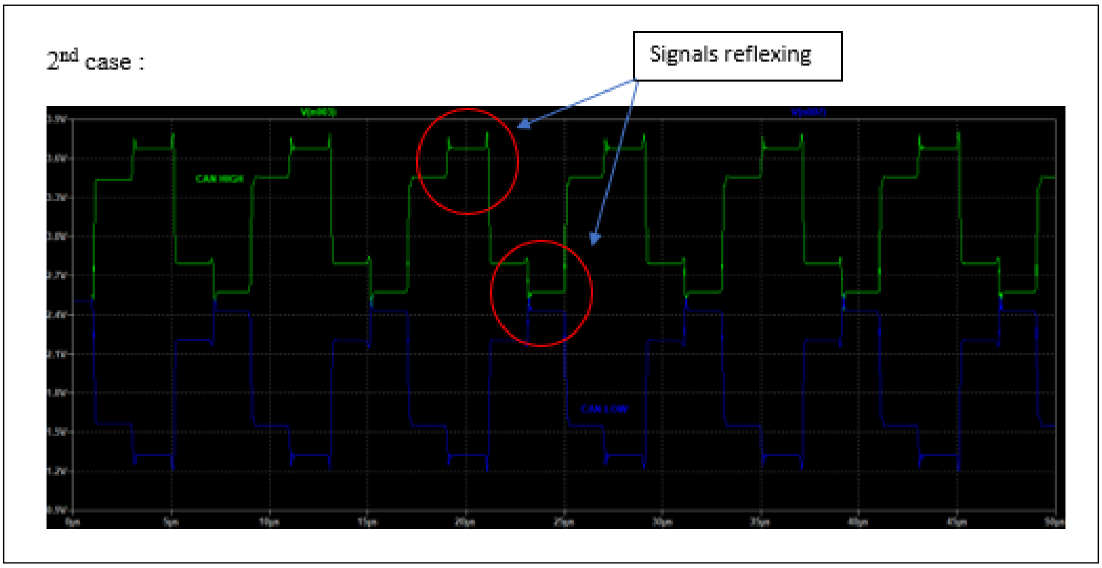

FIGURE 1.40 – 2nd case signal reflection visualization

Waveforms :

The introduction of a stub creates multiple signal paths, causing reflections and timing issues.

Time-Domain Analysis : Observing the delays and distortions caused by the signal reflections.

Observation : The presence of stubs causes the signal to split, resulting in timing mismatches and data corruption. The time-domain analysis shows the impact on the signal waveform.

Details :

Stub Length and Impedance : Different lengths and impedances of the stub are tested to observe their effects on signal reflections.

Timing Analysis : The impact on signal propagation delay and timing jitter is analyzed.

1.6.2.3 Termination Issues

Simulation Setup :

R1 120 Ohm.

R2 60 Ohm.

Visualization :

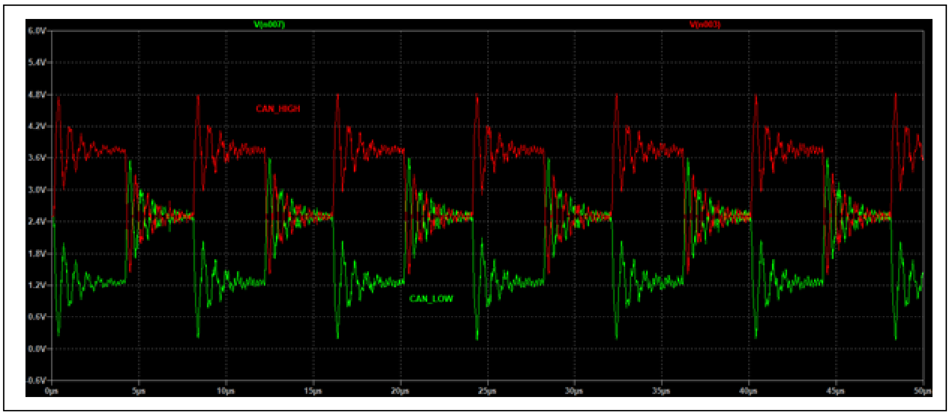

FIGURE 1.41 – Termination issues visualization

Waveforms :

- CANH and CANL signals,in figure 1.41, are observed for increased reflections due to improper termination. **Eye Diagram Analysis** : Eye diagrams are generated to visualize the impact on signal quality.

Observation : Improper termination results in increased reflections, degrading signal quality. The eye diagram analysis shows closure of the eye, indicating potential data errors.

Details :

Termination Resistor Values : Different values of termination resistors are tested to find the optimal termination.

Signal Quality Metrics : Metrics such as bit error rate (BER) and eye opening are used to evaluate the impact of termination issues.

1.6.2.4 Noise Effects

Simulation Setup :

Adding noise to the CAN High Line :

- Vnoise, n001, 0, AC, 0.1m

- Rnoise, n001, CANH, 1k

Adding noise to the CAN Low Line :

- Vnoise2, n002, 0, AC, 0.1m

- Rnoise2, n002, CANL, 1k

Visualization :

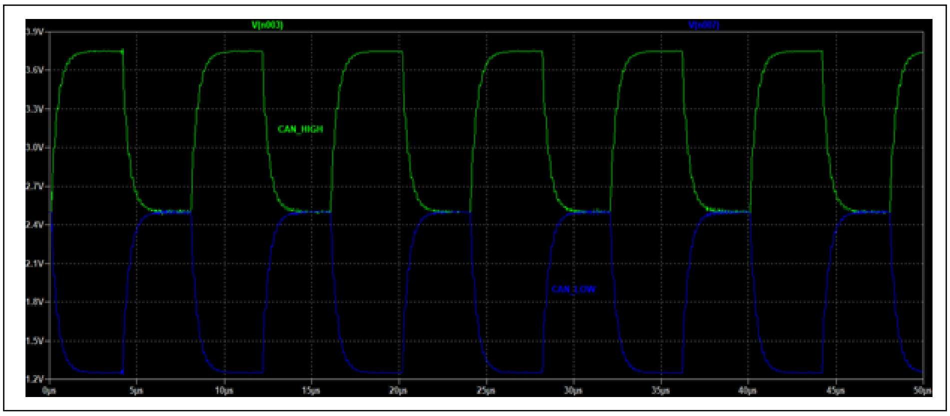

FIGURE 1.42 – Noise effects signal visualization

CAN HIGH visualization :

.

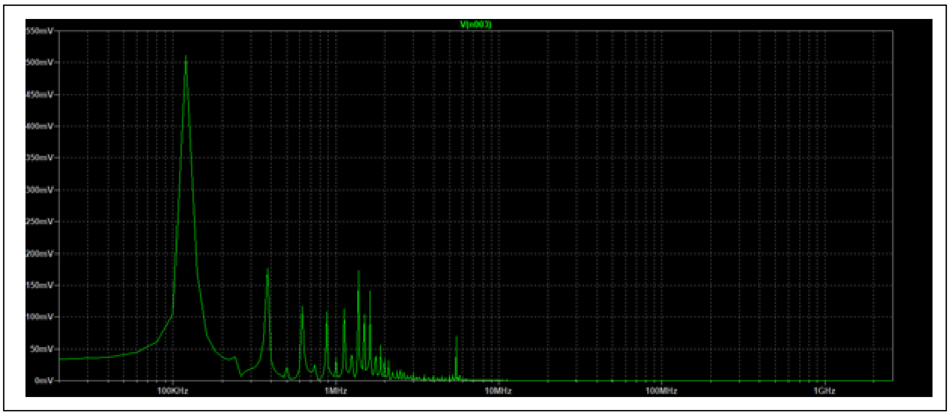

FIGURE 1.43 – The FFT visualization of the CAN High in Linear

.

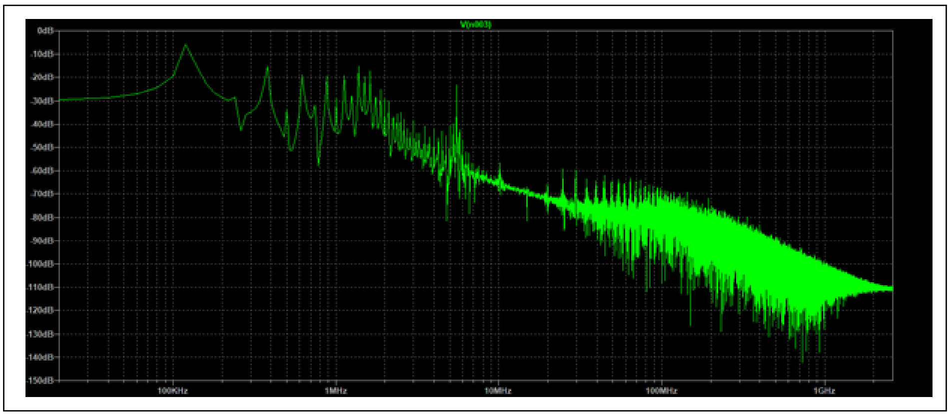

FIGURE 1.44 – The FFT visualization of the CAN High in Decibel

.

CAN LOW visualization :

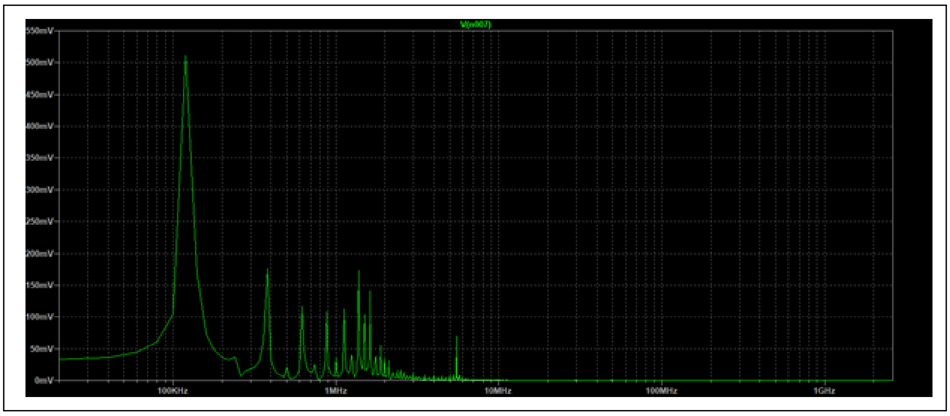

FIGURE 1.45 – The FFT visualization of the CAN Low in linear

.

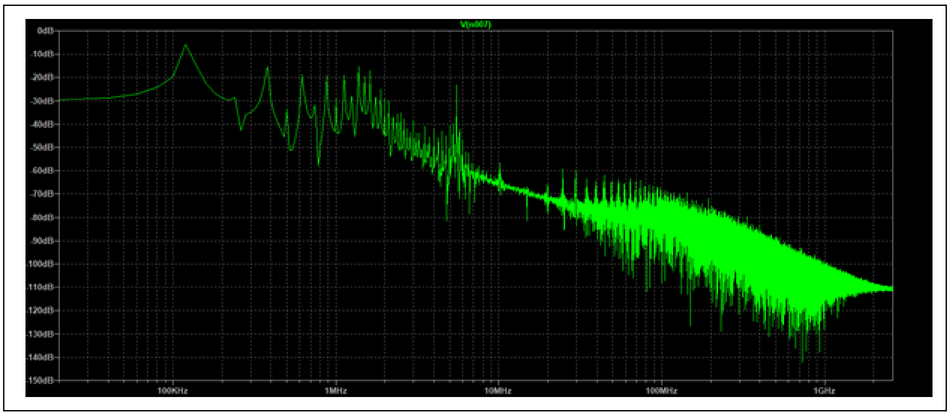

FIGURE 1.46 – The FFT visualization of the CAN Low in Decibel

Waveforms : CANH and CANL signals, in figure 1.42, are observed for variations due to noise.

Frequency-Domain Analysis : FFT is applied to analyze the noise impact on different frequency components.(Figure 1.43 and figure 1.45)

Observation : Noise introduces random variations, which can corrupt data and trigger error frames in the CAN protocol. The frequency-domain analysis shows increased noise components in the signal spectrum.

Details :

Noise Source Characteristics : Different types and levels of noise sources are tested to simulate real-world interference.

Error Analysis : The impact of noise on error rates and signal quality is quantified.

1.6.3 Common Failure Scenarios and Troubleshooting Strategies

1.6.3.1 Bus Off Error

Symptoms :

Occurs when a node transmits too many messages in a short time or encounters a severe fault, leading to the node being taken offline and disrupting communication.

Troubleshooting Strategy :

Error Logs : Check error logs to identify the specific node that triggered the Bus Off error.

Node Inspection : Inspect the node for faults or issues causing excessive message transmission.

Address Underlying Issues : Resolve any underlying problems with the node and reset it.

Network Monitoring : Monitor the network for stability after resolving the issue.

Flowchart :

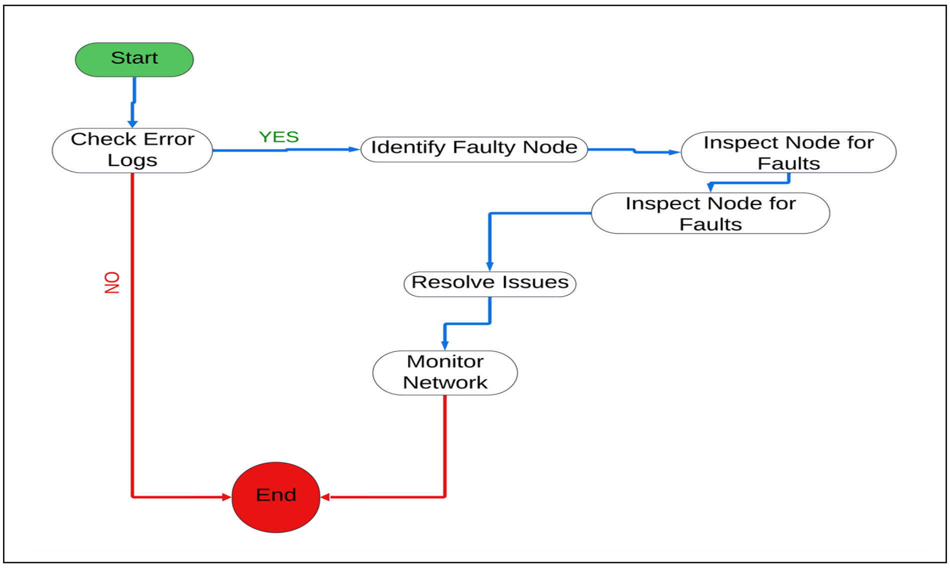

FIGURE 1.47 – CAN Bus Off Error Detection and Resolution Flowchart

1.6.3.2 Excessive Jitter or Delay

.

Symptoms :

Leads to unreliable communication and synchronization issues between nodes.

Troubleshooting Strategy :

Timing Parameters : Review the timing parameters and settings of all nodes.

Bit Rates : Ensure nodes adhere to specified bit rates and synchronization requirements.

Root Cause Investigation : Investigate hardware or software issues causing jitter or delay.

Adjustments : Adjust node configurations or update software to minimize timing discrepancies.

Flowchart :

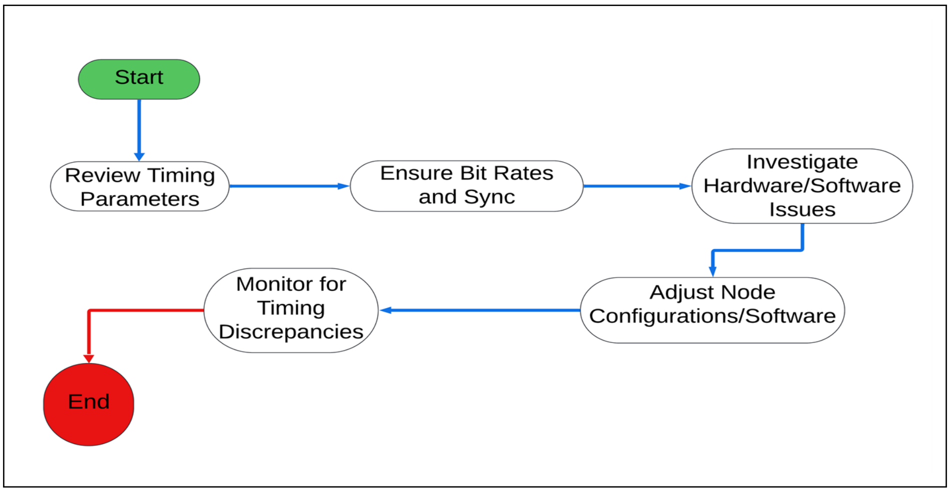

FIGURE 1.48 – CAN Bus Timing Parameter Review Process

1.6.3.3 Message Collision

.

Symptoms :

Occurs when multiple nodes attempt to transmit messages simultaneously, resulting in corrupted data and communication disruptions.

Troubleshooting Strategy :

Arbitration Process : Analyze the message arbitration process to identify nodes involved in collisions.

Unique IDs : Verify nodes are correctly configured with unique message IDs.

Message Priorities : Adjust message priorities and IDs to prevent collisions.

Collision Resolution : Implement a collision resolution strategy, such as message queuing or prioritization.

Flowchart :

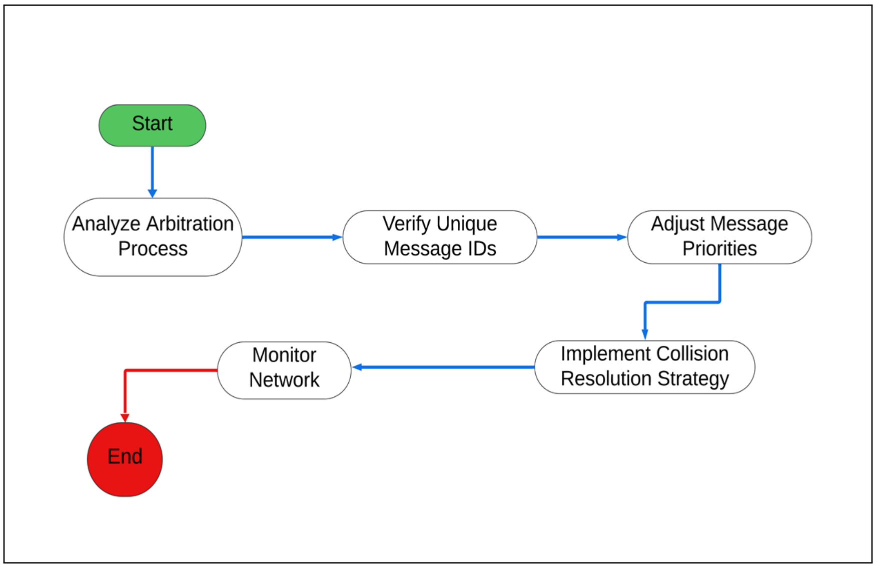

FIGURE 1.49 – CAN Bus Arbitration Process Analysis

1.6.3.4 Signal Integrity Issues

.

Symptoms :

Manifest as distorted or noisy waveforms on the CAN Bus, leading to errors in message reception.

Troubleshooting Strategy :

Physical Layer Inspection : Inspect cables, connectors, and terminations for issues.

Connection Check : Check for loose or damaged connections introducing signal noise.

Termination Resistors : Ensure termination resistors are correctly placed and functioning.

Oscilloscope Analysis : Use an oscilloscope to analyze signal waveforms and identify interference sources.

EMI Minimization : Shield cables or re-route them to minimize electromagnetic interference.

Flowchart :

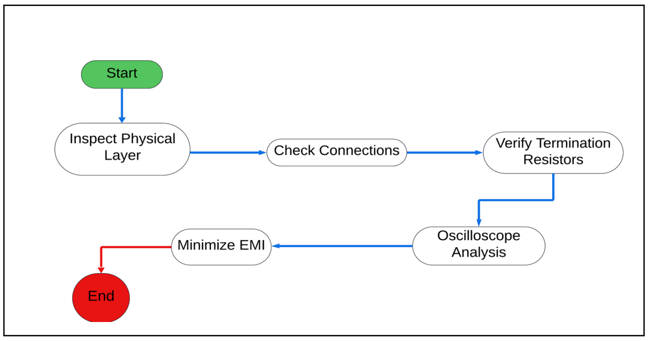

FIGURE 1.50 – CAN Bus Physical Layer Inspection and Verification

1.6.3.5 Node Failures

.

Symptoms :

Individual nodes may fail, leading to a loss of communication with that node and potential system disruptions.

Troubleshooting Strategy :

Isolate Malfunctioning Node : Disconnect the malfunctioning node from the network.

Node Inspection : Perform a thorough inspection for physical damage or component failures.

Software/Firmware Check : Check for software or firmware issues on the node's microcontroller.

Repair/Replace Node : Replace or repair the faulty node and ensure proper configuration before reconnecting.

Post-Repair Monitoring : Monitor the network for stability after addressing the node failure.

Flowchart :

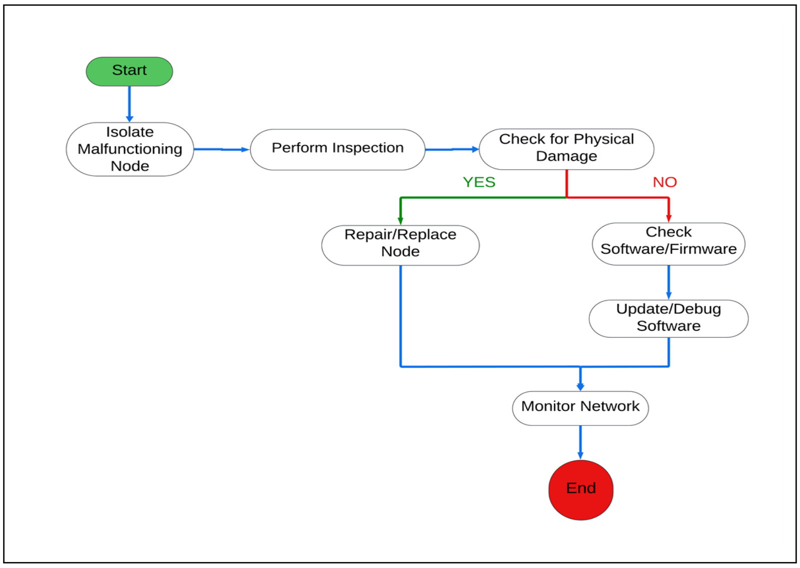

FIGURE 1.51 – **Troubleshooting Malfunctioning CAN Bus Node**

1.6.3.6 EMI and EMC Interference

.

Symptoms :

EMI and EMC issues can disrupt CAN Bus communication, leading to data corruption and errors.

Troubleshooting Strategy :

EMI Sources Evaluation : Identify potential EMI sources such as high-voltage equipment or radio transmitters.

Cable Shielding : Shield CAN Bus cables and connectors to reduce susceptibility to external interference.

Twisted-Pair Cables : Use twisted-pair cables and filters to minimize EMI effects.

EMC Compliance : Ensure the CAN Bus network complies with EMC standards and regulations.

EMC Testing : Conduct electromagnetic compatibility testing to validate the network's immunity to interference.

Flowchart :

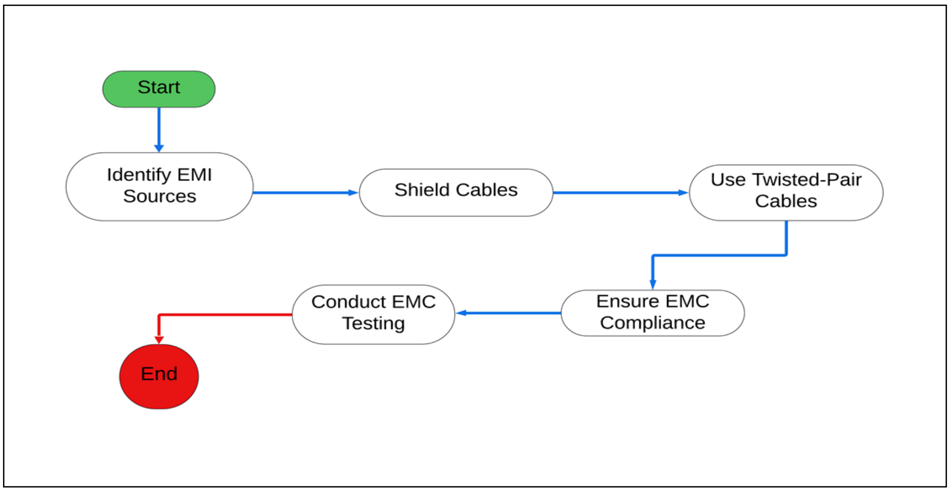

FIGURE 1.52 – CAN Bus Electromagnetic Interference (EMI) Mitigation

1.6.3.7 Software Bugs and Protocol Violations

.

Symptoms :

Software bugs or protocol violations in node firmware or software can cause unexpected behavior and communication errors.

Troubleshooting Strategy :

Code Review : Review the software code of the affected node for bugs or protocol violations.

Debugging : Correct the software to adhere to CAN Bus protocol standards.

Firmware Updates : Verify the node's firmware or software is up to date.

Controlled Testing : Test the corrected software in a controlled environment before deploying it.

Flowchart :

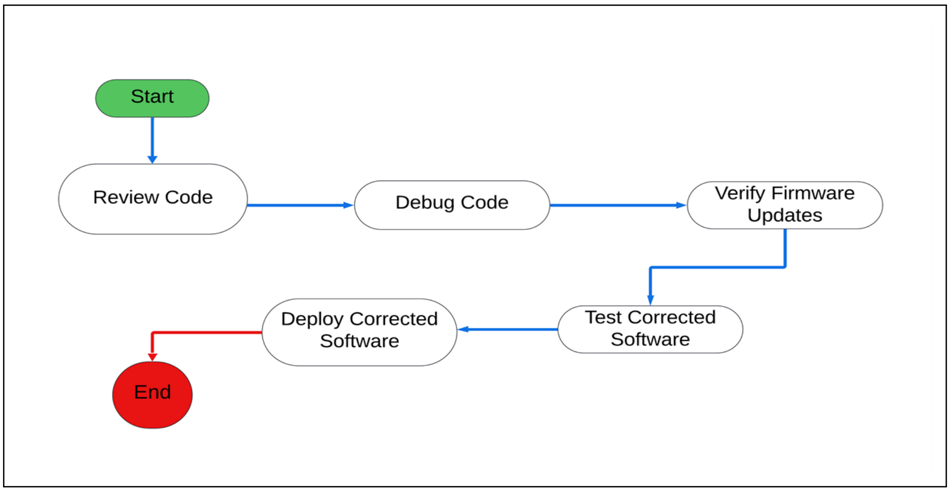

FIGURE 1.53 – CAN Bus Code Review and Debugging Process

1.6.3.8 Power Supply Issues

.

Symptoms :

Inadequate or unstable power supplies can lead to voltage drops or spikes on the CAN Bus, affecting node operation.

Troubleshooting Strategy :

Power Supply Verification : Ensure all nodes receive a stable and sufficient power supply within the specified voltage range.

Power Distribution Inspection : Inspect power distribution circuits and connectors for issues.

Voltage Regulation : Implement voltage regulation and filtering to maintain a consistent power supply.

Power Quality Monitoring : Monitor power quality and voltage levels during network operation to detect irregularities.

Flowchart :

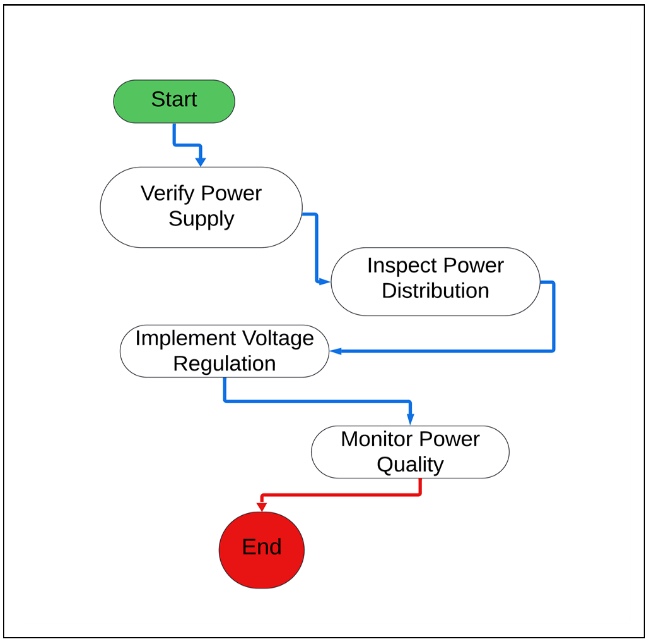

FIGURE 1.54 – Verifying and Implementing Power Supply in CAN Bus Systems

1.6.4 Enhancements and Mitigations

.

1.6.4.1 Proper Termination and Impedance Matching

Diagram : Show a transmission line with consistent impedance values.

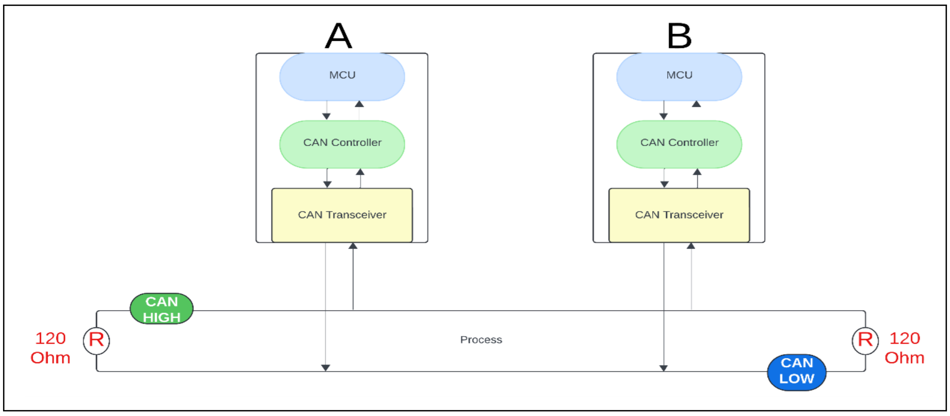

FIGURE 1.55 – Simplified CAN Bus Network with Two Nodes and Termination Resistors

1.6.4.2 Noise Reduction and Stub Minimization

.

The difference in noise immunity using twisted-pair cables.

Feature	Shielded Cables	Unshielded Cables
Construction	Additional shielding layer (foil, braided)	Insulated conductors twisted together
Purpose	Protects against EMI and RFI	Basic signal transmission
Advantages	Better EMI/RFI protection / Reduced crosstalk / Suitable for high-frequency applications	Cost-effective / Lighter and more flexible / Easy installation
Disadvantages	Higher cost / Thicker and less flexible / Weight and bulk	Susceptible to EMI/RFI / Increased crosstalk / Limited use in high-interference areas
Common Uses	Industrial networking, aerospace / Telecommunications, medical equipment	Residential and commercial networking / Basic audio/video connections

TABLE 1.4 – Cables type Comparison

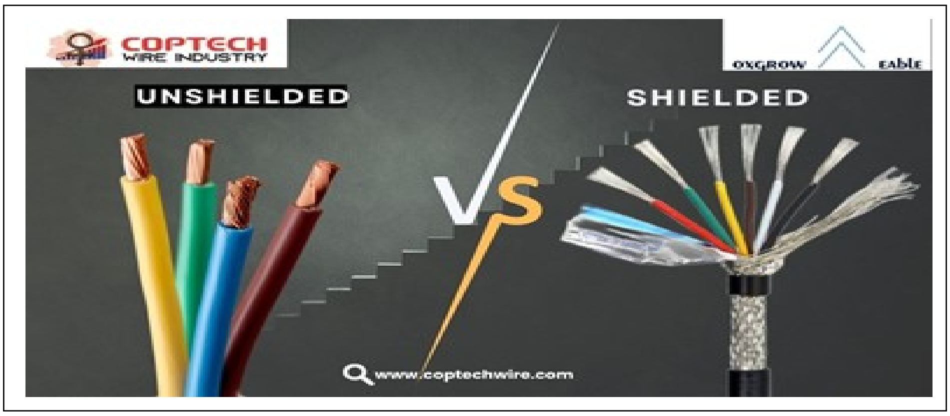

FIGURE 1.56 – CAN Bus Network with Three Nodes and Termination Resistors

The difference in noise immunity using twisted-pair cables.

Feature	Twisted-Pair Cable without Shielding	Twisted-Pair Cable with Shielding
Construction	Two insulated copper wires twisted together	Two insulated copper wires twisted together, with additional shielding (foil, braided mesh, or both)
EMI Protection	Susceptible to external electromagnetic interference (EMI)	Provides enhanced protection against external EMI
Crosstalk	Moderate crosstalk between adjacent pairs	Reduced crosstalk between adjacent pairs
Applications	General data transmission, telephony	Networking, telecommunications, industrial environments
Advantages	Cost-effective, lightweight	Improved noise immunity, better signal integrity
Disadvantages	Limited protection against EMI and crosstalk	Higher cost, thicker and less flexible

TABLE 1.5 – Twisted-Pair Cable type Comparison

***Shielding** : Adding shielding further enhances the cable's ability to reject external interference, making it suitable for critical applications in networking and telecommunications.

1.6.4.3 Component Quality and Testing

Flowchart :

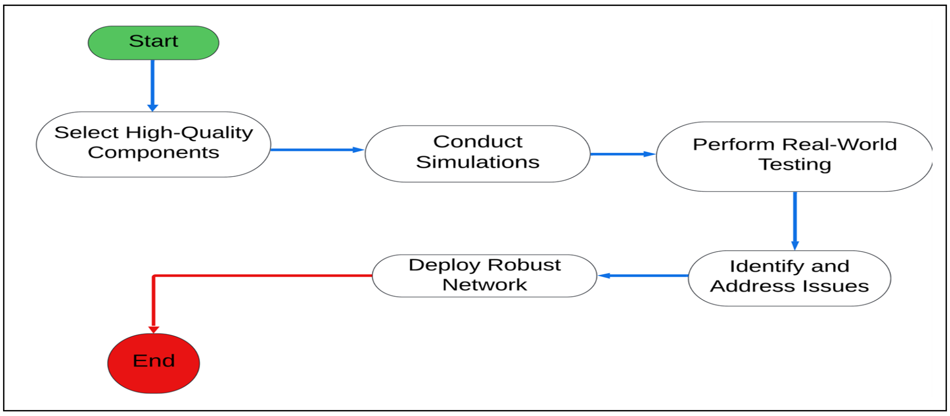

FIGURE 1.57 – Ensuring Robust CAN Bus Network Deployment

1.6.4.4 Example Visualizations

.

Flowchart for Node Failures

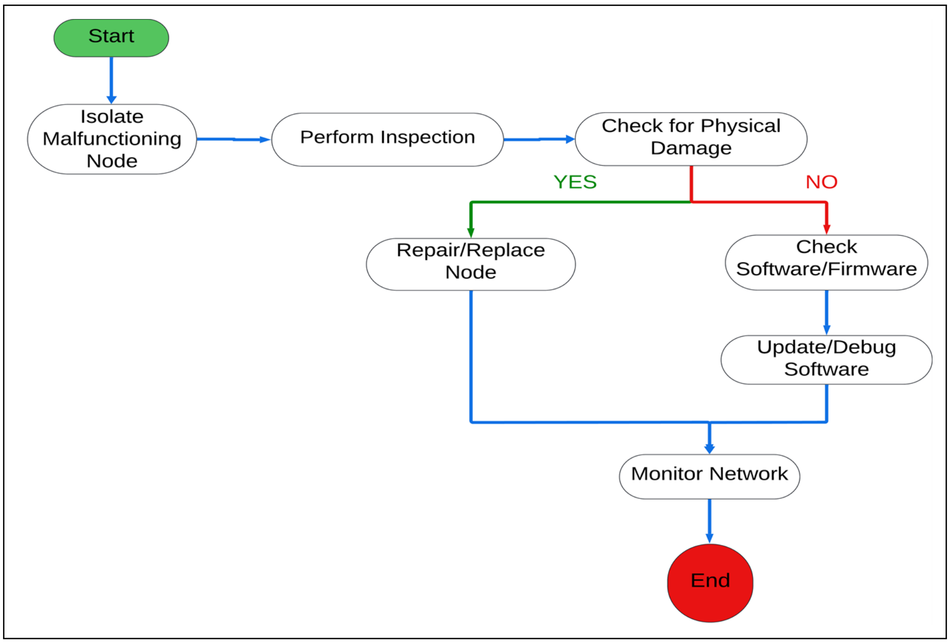

FIGURE 1.58 – CAN Bus Node Failure strategy

.

1.6.4.5 Network Diagram with Problematic Node

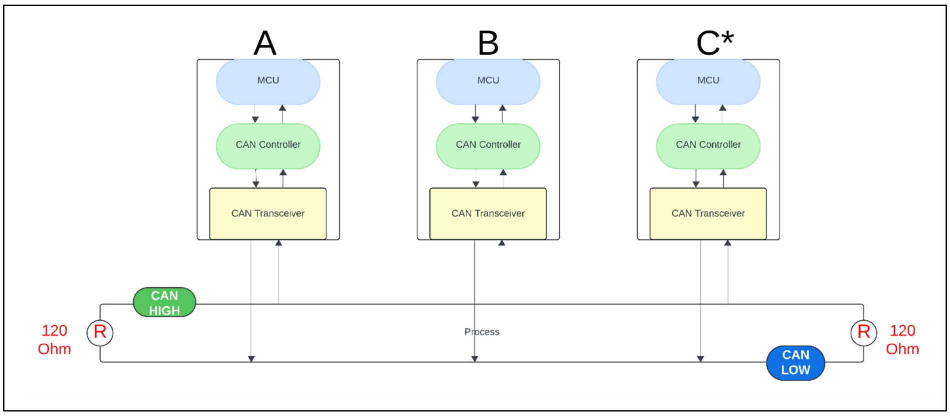

FIGURE 1.59 – CAN Bus Network with Three Nodes and Termination Resistors

* Node C identified as problematic

1.7 The KIT AURIX TC265 TFT

The KIT AURIX TC265DE is a versatile development platform designed around the Infineon AURIX TC26 TFTmicrocontroller, tailored for advanced automotive and industrial applications. This strategic report explores its key features, market positioning, and strategic advantages.

1.7.1 Key Features and Capabilities

The KIT AURIX TC265 TFT leverages the following core features :

• **Microcontroller Powerhouse** : Powered by the AURIX TC26xD, featuring dual 32-bit TriCore CPUs running at up to 200 MHz. It boasts up to 2.5 MB of Flash memory and 240 KB of RAM, ideal for handling complex computations and real-time processing.

• **Display and Storage** : Equipped with an XGA LCD display (320x240 pixels) and a microSD card slot, enabling advanced graphical user interfaces and robust data storage capabilities crucial for HMI applications.

• **Communication Interfaces** : Includes Ethernet PHY for high-speed networking, LIN transceiver for automotive field buses, and high-speed CAN transceiver for reliable and real-time control networks. These interfaces facilitate seamless integration into complex systems requiring robust communication protocols.

• **Additional Components** : Features such as a Real-Time Clock (RTC) with alarm, SRAM, and a unique MAC ID enhance time-sensitive data management. It also includes a micro-USB connector, buzzer, LEDs, and push-button for versatile user interaction and system monitoring.

FIGURE 1.60 – The KIT AURIX TC265 TFT

1.7.2 Market Positioning

The KIT AURIX TC265 TFT is strategically positioned in the following markets :

• **Automotive Industry** : Positioned for critical automotive applications requiring high reliability, performance, and robust communication capabilities. Applications include engine control, chassis control, and advanced driver assistance systems (ADAS).

• **Industrial Automation** : Suited for industrial automation applications demanding real-time data processing, secure communication, and scalability. It finds applications in robotics, industrial IoT (IIoT), and smart factory environments.

• **Emerging Technologies** : Supports emerging technologies such as smart cities, autonomous vehicles, and renewable energy systems where reliability, performance, and connectivity are paramount.

1.7.3 Strategic Advantages

The strategic advantages of the KIT AURIX TC265 TFT include :

• **High Performance and Reliability** : Offers dual TriCore CPUs and extensive memory capabilities, ensuring robust performance in demanding environments.

• **Versatile Connectivity** : Provides a wide range of communication interfaces, enabling integration into diverse systems and networks.

• **Scalability and Flexibility** : Modular architecture allows customization and expansion to meet evolving application requirements, enhancing long-term viability and adaptability.

• **Developer-Friendly Environment** : Supported by a comprehensive software development ecosystem including IDEs and toolchains, facilitating rapid prototyping and efficient development cycles.

1.7.4 CAN in KIT AURIX TC265 TFT

The KIT AURIX TC265 TFT board does indeed integrate a high-speed CAN transceiver to enable communication via the CAN bus. Specifically :

- The CAN0 component on the board is a high-speed CAN transceiver.
- It is connected to a 10-pin header (2x5) on the board to facilitate CAN connections.
- The AURIX TC26xD microcontroller at the core of the board manages CAN communications via this CAN0 transceiver.

=> Therefore, to communicate through a CAN bus, the KIT AURIX TC265 TFT board does have the necessary CAN transceiver integrated directly on the board. This allows the platform to easily interface with other nodes on a CAN network, which is essential for automotive and industrial applications.

Transceiver :

High-Speed CAN Transceiver : Infineon TLE6250 G

LIN Transceiver : Infineon TLE7259-2GE

Ethernet Gigabit PHY : PEF7071 (LANTIQ)

CAN Controller :

The CAN Controller is integrated into the main microcontroller, Infineon AURIX TC265.

Micro-Controller Unit (MCU)

Microcontroller : Infineon AURIX TC265 :

Architecture : TriCore™

Cores : Two 32-bit TriCore cores running up to 200 MHz

Memory : Up to 2.5 MB of flash memory and 240 KB of RAM

Package : LQFP-176

These components are essential for communication, control, and data processing in advanced automotive and industrial applications.

FIGURE 1.61 – The main components of the Kit AURIX TC265 TFT

1.8 CONCLUSION

In conclusion, this theoretical study highlights the essential role of proper impedance matching, termination, and noise mitigation in preserving the signal integrity of CAN bus communication. Our simulations have clearly demonstrated that adaptation issues and noise can lead to significant degradation in communication reliability, emphasizing the need for meticulous design and implementation in practical applications. The temporal analysis of the CAN bus, supported by robust error handling mechanisms, affirms the protocol's reliability and effectiveness in embedded systems. With an error detection capability that ensures less than one detected error per 1000 years of operation, the CAN bus stands out as a highly reliable communication solution. Its resilience in quickly recovering from interruptions and maintaining high performance underscores its suitability for a wide range of embedded applications, ensuring minimal disturbances and maximum operational efficiency.