GROUP 5

PRINCE MISHRA :- What is CAN
USHA :- Why use CAN
JYOTISMAN KIRTI PRAKASH :- Key characteristics of CAN
DEEPAK :- How CAN works
SUMA :- Android Auto Content

Understanding the CAN Bus

The Controller Area Network (CAN) bus is a robust vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer. It is a message-based protocol, meaning that messages are not addressed to specific stations but are broadcast to all nodes on the network.

Why use CAN?

Main Reasons We Use CAN

1. Reduces Wiring Complexity

- Without CAN: every sensor needs a dedicated connection to every ECU that needs its data → a wiring nightmare.
- With CAN: all devices connect to the **same two wires (CAN_H & CAN_L)**.
- **Result** → Lighter, cheaper, and easier to maintain wiring harness.

---

2. Real-Time Communication

- CAN is **fast enough** for most automotive control systems (up to 1 Mbps for Classical CAN, higher for CAN FD).
- Priority-based **message arbitration** means critical signals (e.g., brake commands) always get through first.

---

3. Multi-Master & Broadcast

- **Multi-master** → Any ECU can start communication, no central controller required.
- **Broadcast nature** → One message can be received by all ECUs at the same time without duplication.

---

4. Robustness & Reliability

- **Differential signaling** resists noise — important in noisy automotive and industrial environments.
- **Error detection** and **automatic retransmission** ensure data integrity.
- Faulty nodes can **go offline automatically** to avoid disturbing the network.

---

5. Cost-Effective

- Fewer cables = lower production cost.
- No expensive central computer required — each ECU is independent but can still share data.

---

6. Standardization

- CAN is an **ISO standard (ISO 11898)**, widely supported by automotive, industrial automation, and robotics systems.
- Compatible with many development tools, chips, and diagnostic devices.

Key Characteristics of CAN

- **Broadcast Communication:** All nodes on the CAN bus receive all messages, allowing for flexible and decentralized communication.
- **Message Prioritization:** Messages have an identifier that also serves as a priority. Lower identifier values have higher priority.
- **Error Detection and Fault Confinement:** CAN has built-in mechanisms for detecting errors and isolating faulty nodes, ensuring data integrity.
- **Multi-Master Capability:** Any node can initiate communication when the bus is free.
- **Non-Destructive Bitwise Arbitration:** When multiple nodes attempt to transmit simultaneously, a non-destructive arbitration process ensures that the highest priority message is transmitted successfully without data loss.

How CAN works ?

1. Purpose of CAN

CAN is a **multi-master, message-based communication protocol** designed for real-time communication between multiple Electronic Control Units (ECUs) in vehicles and industrial systems — without needing a central computer.
It’s:

- **Robust** (handles noise & faults)
- **Efficient** (small messages, priority-based)
- **Deterministic** (high-priority messages always win arbitration)

---

2. CAN Bus Basics

- **Two-wire system**: `CAN_H` (high) & `CAN_L` (low)
- **Differential signaling**: Improves noise immunity
    - **Dominant bit (0)**: CAN_H voltage > CAN_L voltage
    - **Recessive bit (1)**: CAN_H ≈ CAN_L
- **Speed**: Typically up to 1 Mbps (Classical CAN), up to 5–8 Mbps (CAN FD)

---

3. Step-by-Step Working of CAN

Step 1 – Data Creation

- A **sensor** or ECU generates data.
- Example: The ABS ECU reads wheel speed and needs to share it with the engine and dashboard ECUs.

Step 2 – CAN Frame Creation

- The data is packaged into a **CAN Frame**:
    - **Identifier (ID)** → Defines message priority (lower ID = higher priority).
    - **Data field** → The actual data (up to 8 bytes in Classical CAN, 64 bytes in CAN FD).
    - **Control bits** → Define frame length/type.
    - **CRC** → For error detection.

Step 3 – Arbitration (Who Speaks First)

- Multiple ECUs may want to send messages **at the same time**.
- CAN uses **CSMA/CA** (Carrier Sense Multiple Access with Collision Avoidance) + **bitwise arbitration**:
    1. All nodes listen to the bus.
    2. If the bus is idle, they start sending.
    3. They send bits and check the bus at the same time.
    4. If a node sends a “1” but reads a “0” → It stops and waits.
    5. **Lowest numerical ID wins**.

Step 4 – Bit Stuffing

- To keep clock synchronization, after **5 consecutive identical bits**, the sender inserts an **opposite bit**.
- This prevents long runs of identical bits.

Step 5 – Physical Transmission

- Data bits are sent as voltage differences between CAN_H & CAN_L.
- All nodes **receive the same message simultaneously**.

Step 6 – Reception by All Nodes

- Every ECU reads the message.
- Hardware filters in each ECU check the **message ID**:
    - If relevant → Process it.
    - If not → Ignore it.

Step 7 – Error Checking

- CAN has 5 error detection methods:

    - **Bit Monitoring**
    - **Bit Stuffing Check**
    - **CRC Check**
    - **ACK Check**
    - **Form Check**
- If any error is found:
    - An **Error Frame** is sent.
    - The faulty message is **discarded** and **automatically retransmitted**.

Step 8 – Data Processing

- The target ECU decodes the payload.
- Sends it to its application logic (e.g., speedometer display, torque control).

---

4. Key Features of CAN

- **Multi-master** → Any ECU can start transmission.
- **Priority-based** → Critical messages get sent first.
- **Fault-tolerant** → Detects and recovers from errors.
- **Broadcast nature** → All ECUs hear the same message

Android Auto Content

What is Android Auto?

Android Auto is Google’s **in-car infotainment platform** that mirrors your Android phone’s apps and functions onto your car’s display, letting you **stay connected, entertained, and informed while keeping your hands on the wheel and eyes on the road**.

It’s either:

- **Wired** → Connected via USB cable.
- **Wireless** → Over Wi-Fi & Bluetooth (supported cars/phones only).

Key Features

1. Navigation & Maps

- **Google Maps** and **Waze** with live traffic updates.
- Voice-guided turn-by-turn navigation.
- Real-time rerouting to avoid delays.

2. Hands-Free Communication

- Make/receive calls and messages with **Google Assistant**.
- Read incoming texts aloud and reply by voice.

3. Music & Audio Streaming

- Spotify, YouTube Music, Amazon Music, and more.
- Podcasts and audiobooks supported.

4. Voice Commands

- Triggered by **"Hey Google"** or steering-wheel button.
- Controls navigation, calls, music, and smart home devices.

5. Third-Party App Support

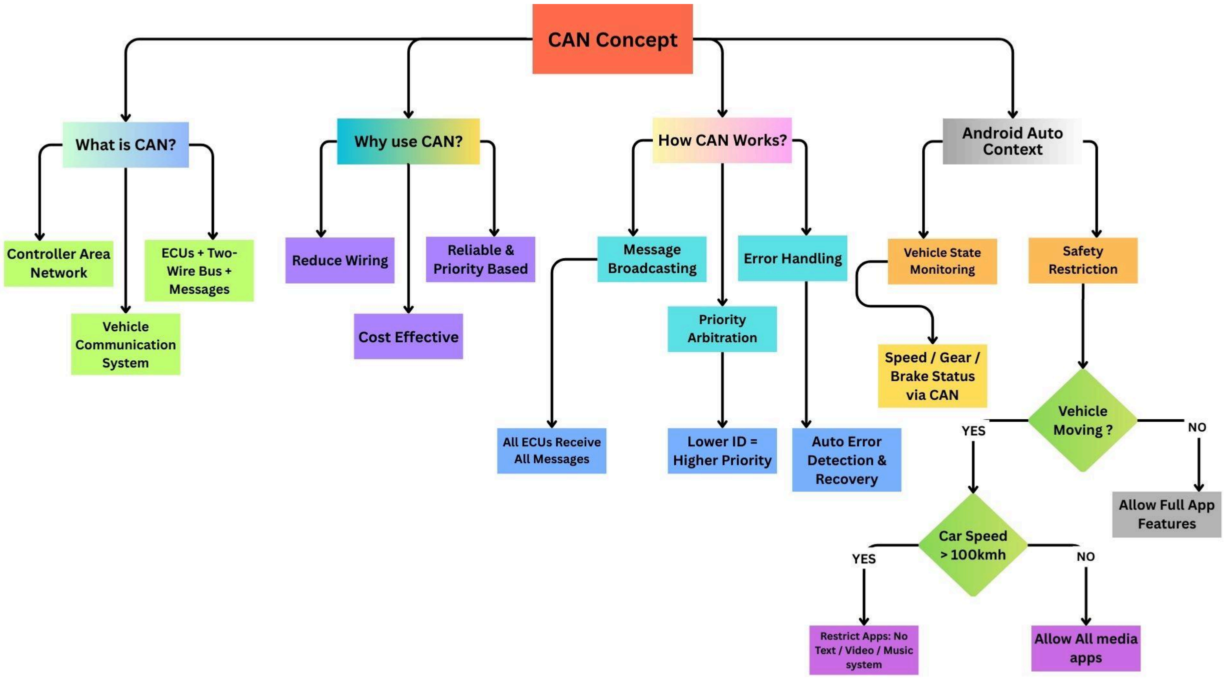
- Messaging: WhatsApp, Telegram.
- Music: Pandora, Tidal.
- Navigation: Sygic, TomTom.

Advantages

- **Safety first** → Minimized distractions via voice and large touch targets.

- **Familiar interface** → Works just like your Android phone.
- **Automatic updates** → New features via phone's Android Auto app updates.
- **Cross-app integration** → Use multiple apps simultaneously (e.g., maps + music).
-

A Detailed Flowchart


CAN Concept
What is CAN?
Controller Area Network
Vehicle Communication System
ECUs + Two-Wire Bus + Messages
Why use CAN?
Reduce Wiring
Cost Effective
Reliable & Priority Based
How CAN Works?
Message Broadcasting
All ECUs Receive All Messages
Priority Arbitration
Lower ID = Higher Priority
Error Handling
Auto Error Detection & Recovery
Android Auto Context
Vehicle State Monitoring
Speed / Gear / Brake Status via CAN
Safety Restriction
Vehicle Moving ?
YES
NO
Allow Full App Features
Car Speed > 100kmh
YES
NO
Restrict Apps: No Text / Video / Music system
Allow All media apps