Controller Area Network

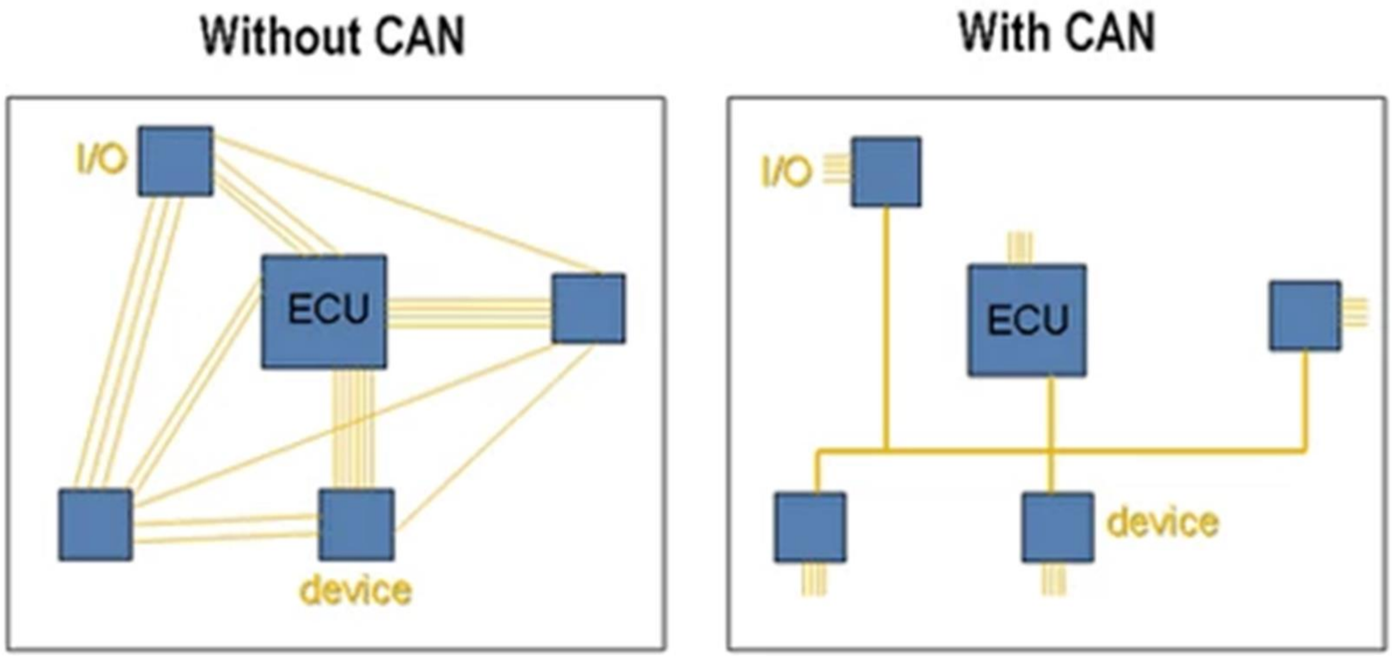
S. Venkatesan

Acknowledgement: The contents, example scripts and some figures are copied from various sources.
Thanks to all authors and sources made those contents public and usable for educational purpose

Controller Area Network Bus

- It is a vehicle bus
  - A **vehicle bus** is a specialized internal communication network that interconnects components inside a vehicle (e.g., automobile, bus, train, industrial or agricultural vehicle, ship, or aircraft).
- Message based Protocol developed in around 1986.
- For each device, the data in a frame is transmitted serially but in such a way that if more than one device transmits at the same time, the highest priority device can continue while the others back off.
- Frames are received by all devices, including by the transmitting device.
- Components that is electronic control units (ECUs) are connected through the CAN bus

Connectivity



Electronic Control Units

- It can be the engine control unit, airbags, audio system etc.
- A modern car may have **up to 70 ECUs** - and each of them may have information that needs to be shared with other parts of the network.
- The CAN bus system enables each ECU to communicate with all other ECUs - without complex dedicated wiring.
- An ECU can prepare and broadcast information (e.g. sensor data) via the CAN bus (consisting of two wires, CAN low and CAN high).
- The broadcasted data is accepted by all other ECUs on the CAN network - and each ECU can then check the data and decide whether to receive or ignore it.

Reasons for the failure of ECU

- Extreme temperature
- Aging
- Alterations

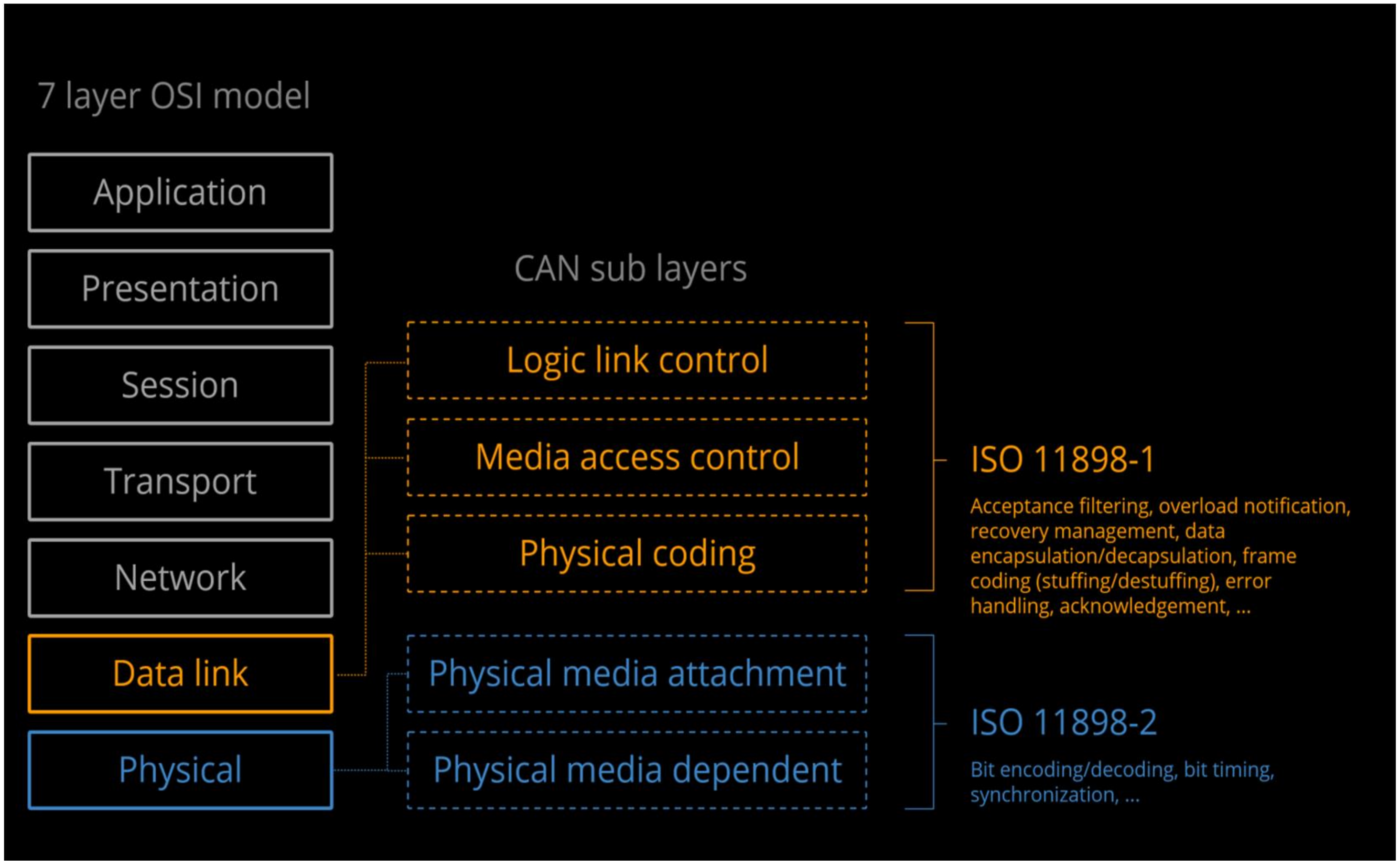
Communications

- **Baud rate:** CAN nodes must be connected via a two wire bus with baud rates up to 1 Mbit/s (Classical CAN) or 5 Mbit/s (CAN FD – Flexible Data Rate)

- **Cable length:** Maximal CAN cable lengths should be between 500 meters (125 kbit/s) and 40 meters (1 Mbit/s)

- **Termination:** The CAN bus must be properly terminated using a 120 Ohms. CAN bus termination resistor at each end of the bus

Layers


7 layer OSI model
Application
Presentation
Session
Transport
Network
Data link
Physical
CAN sub layers
Logic link control
Media access control
Physical coding
Physical media attachment
Physical media dependent
ISO 11898-1
Acceptance filtering, overload notification, recovery management, data encapsulation/decapsulation, frame coding (stuffing/destuffing), error handling, acknowledgement, ...
ISO 11898-2
Bit encoding/decoding, bit timing, synchronization, ...

CAN Bus Packet Format

- SOF and EOF – Start and End of Frame
- SRR–The substitute remote request (SRR) bit replaces the RTR bit in the standard message location as a placeholder in the extended format.
- IDE–A recessive bit in the identifier extension (IDE) indicates that more identifier bits follow. The 18-bit extension follows IDE
- RTR - Remote Transit Request, which is set to dominant if requesting info from another node.
- DLC - Data length code is a half byte long and tells how long the actual data being sent in the data field is.

Priority

- The value of the CAN identifier (CAN-ID) indicates the priority level.
- The lower the number, the higher the priority.
- The ID relates to specific items and activities, such as switching lights on or off, or a particular sensor.

CAN database

- A CAN DBC file (CAN database) is a text file that contains information for decoding raw CAN bus data to 'physical values'.

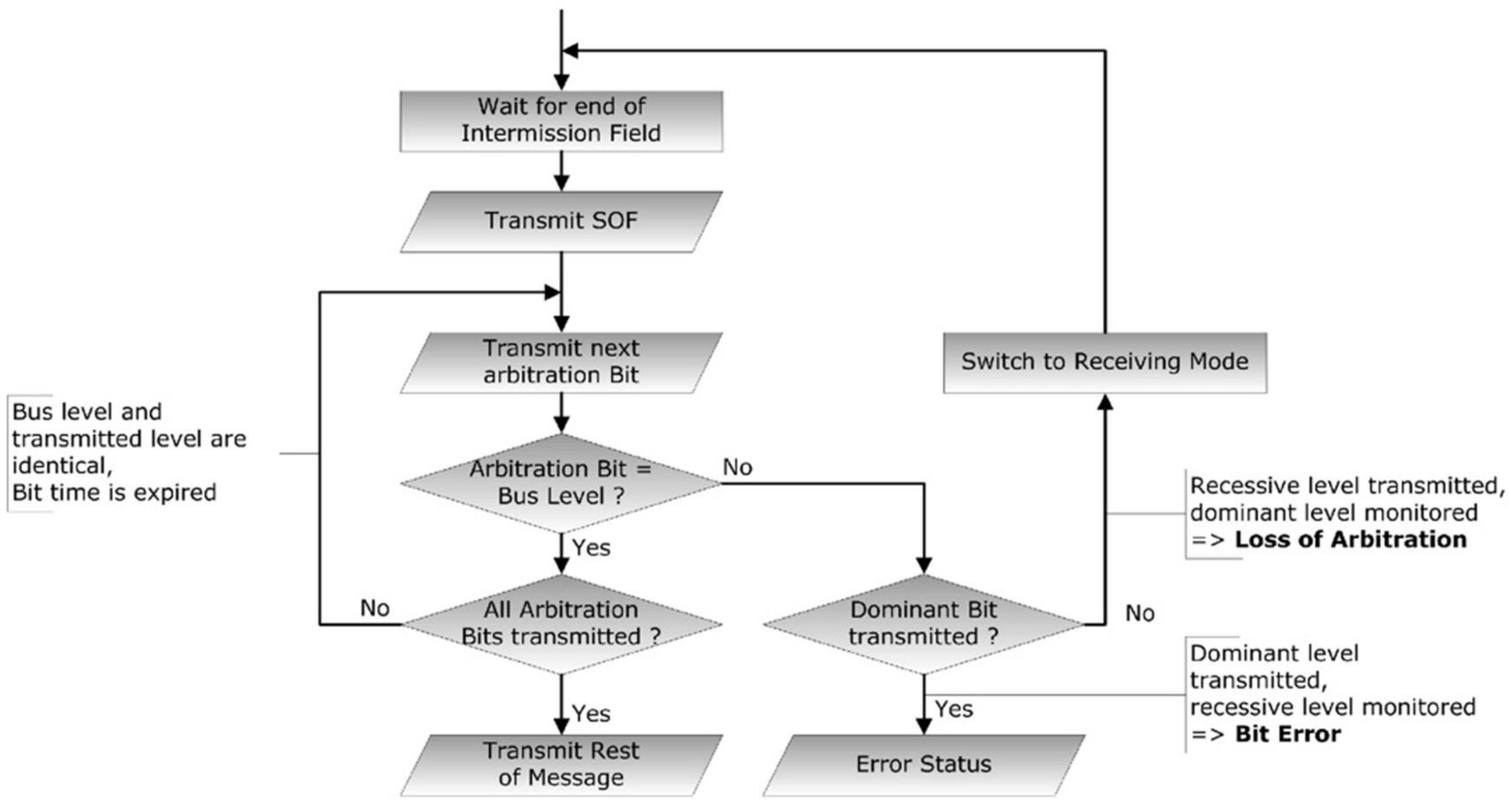
```
CAN ID    Data bytes
0CF00400  FF FF FF 68 13 FF FF FF
```

```
Message  Signal       Value  Unit
EEC1     EngineSpeed  621    rpm
```

CAN Communication

- CAN is a peer-to-peer network. This means that there is no master that controls when individual nodes have access to read and write data on the CAN bus.
- When a CAN node is ready to transmit data, it checks to see if the bus is busy and then simply writes a CAN frame onto the network.
- The CAN frames that are transmitted do not contain addresses of either the transmitting node or any of the intended receiving node(s).
- Instead, an arbitration ID that is unique throughout the network labels the frame.
- All nodes on the CAN network receive the CAN frame, and, depending on the arbitration ID of that transmitted frame, each CAN node on the network decides whether to accept the frame.

Transmission Check



Picture 6.2.2: Arbitration Process Flow Chart

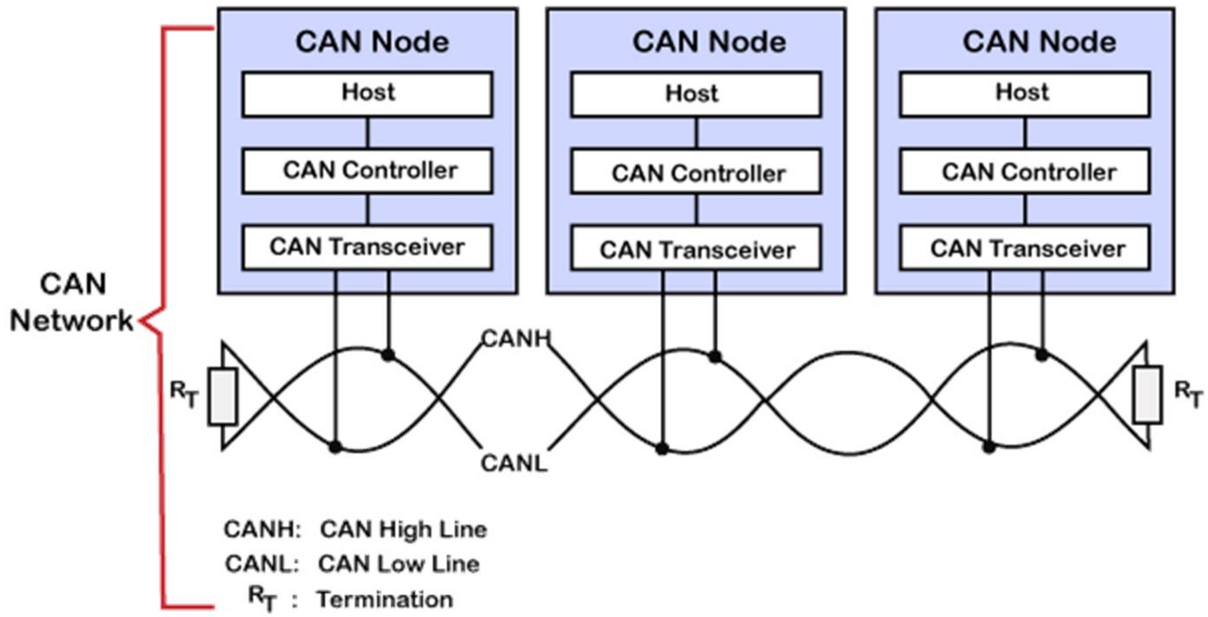
Collision Avoidance

- If multiple nodes try to transmit a message onto the CAN bus at the same time, the node with the highest priority (lowest arbitration ID) automatically gets bus access.

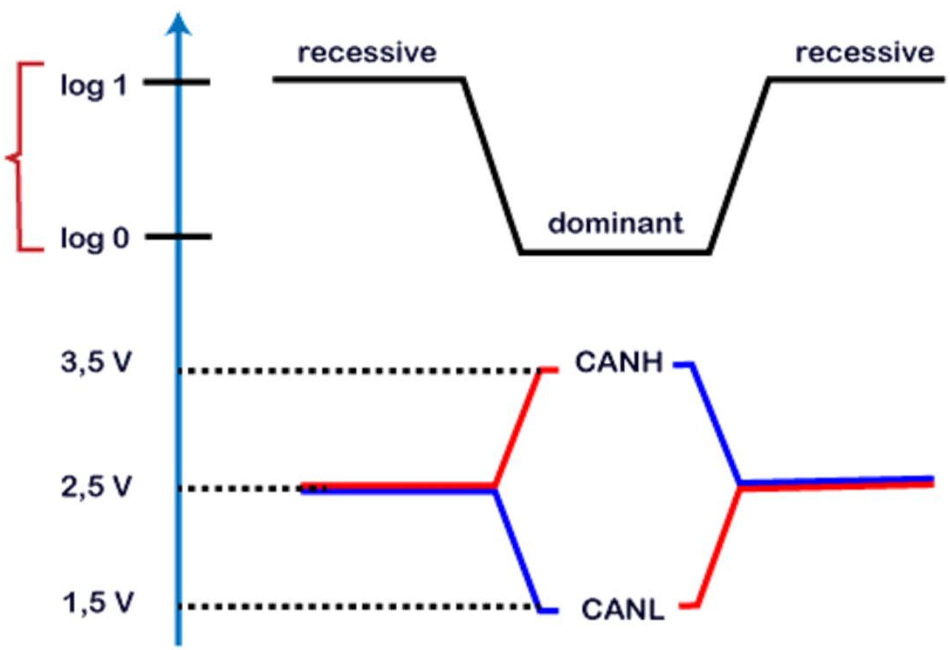
- Lower-priority nodes must wait until the bus becomes available before trying to transmit again.

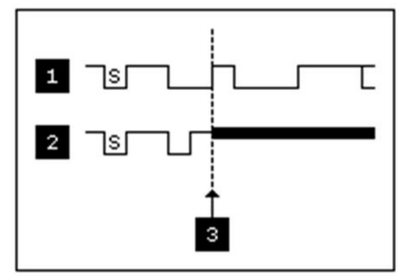
- In this way, you can implement CAN networks to ensure deterministic communication among CAN nodes.

Node Connection



Signal Level





1 Device A: ID = 11001000111 (647 hex)
2 Device B: ID = 11011111111 (6FF hex)
3 Device B Loses Arbitration; Device A Wins Arbitration and Proceeds
S = Start Frame Bit

Figure 5. CAN contains built in priority for messages to avoid conflicts.

Bus Signal

- Logic 1 is a recessive state. To transmit 1 on CAN bus, both CAN high and CAN low should be applied with 2.5V.

- Logic 0 is a dominant state. To transmit 0 on CAN bus, CAN high should be applied at 3.5V and CAN low should be applied at 1.5V.

- The ideal state of the bus is recessive.

- If the node reaches the dominant state, it cannot move back to the recessive state by any other node.

- When CAN high line and CAN low line are applied with 2.5 volts, then the actual differential voltage would be zero volt.

Error Message

- Reasons: Faulty cables, noise, incorrect termination, malfunctioning CAN nodes etc.
- when CAN node 1 detects an error during the transmission of a CAN message, it immediately transmits a sequence of 6 bits of the same logic level - also referred to as raising an Active Error Flag.
- **Bit Error:** Every CAN node on the CAN bus will monitor the signal level at any given time - which means that a transmitting CAN node also "reads back" every bit it transmits. If the transmitter reads a different data bit level vs. what it transmitted, the **transmitter** detects this as a Bit Error.
- **Bit Stuffing Error:** As explained, bit stuffing is part of the CAN standard. It dictates that after every 5 consecutive bits of the same logical level, the 6th bit must be a complement. This is required to ensure the on-going synchronization of the network by providing rising edges. Further, it ensures that a stream of bits are not misinterpreted as an error frame or as the interframe space (7 bit recessive sequence) that marks the end of a message.
  - All CAN nodes automatically remove the extra bits.
- **Form Error:** This message-level check utilises the fact that certain fields/bits in the CAN message must always be of a certain logical level. Specifically the 1-bit SOF must be dominant, while the entire 8-bit EOF field must be recessive. Further, the ACK and CRC delimiters must be recessive. If a receiver finds that any of these are bits are of an invalid logical level, the **receiver** detects this as a Form Error.

Error Message

- **ACK Error (Acknowledgement):** When a transmitter sends a CAN message, it will contain the ACK field (Acknowledgement), in which the transmitter will transmit a recessive bit. All listening CAN nodes are expected to send a dominant bit in this field to verify the reception of the message (regardless of whether the nodes are interested in the message or not). If the transmitter does not read a dominant bit in the ACK slot, the **transmitter** detects this as an ACK Error.

- **CRC Error (Cyclic Redundancy Check):** Every CAN message contains a Cyclic Redundancy Checksum field of 15 bits. Here, the transmitter has calculated the CRC value and added it to the message. Every receiving node will also calculate the CRC on their own. If the receiver's CRC calculation does not match the transmitter's CRC, the **receiver** detects this as a CRC Error.

- Active Error Flag six dominant bits – Transmitted by a node detecting an error on the network that is in error state *error active*.

- Passive Error Flag six recessive bits – Transmitted by a node detecting an active error frame on the network that is in error state *error passive*.

Advantages

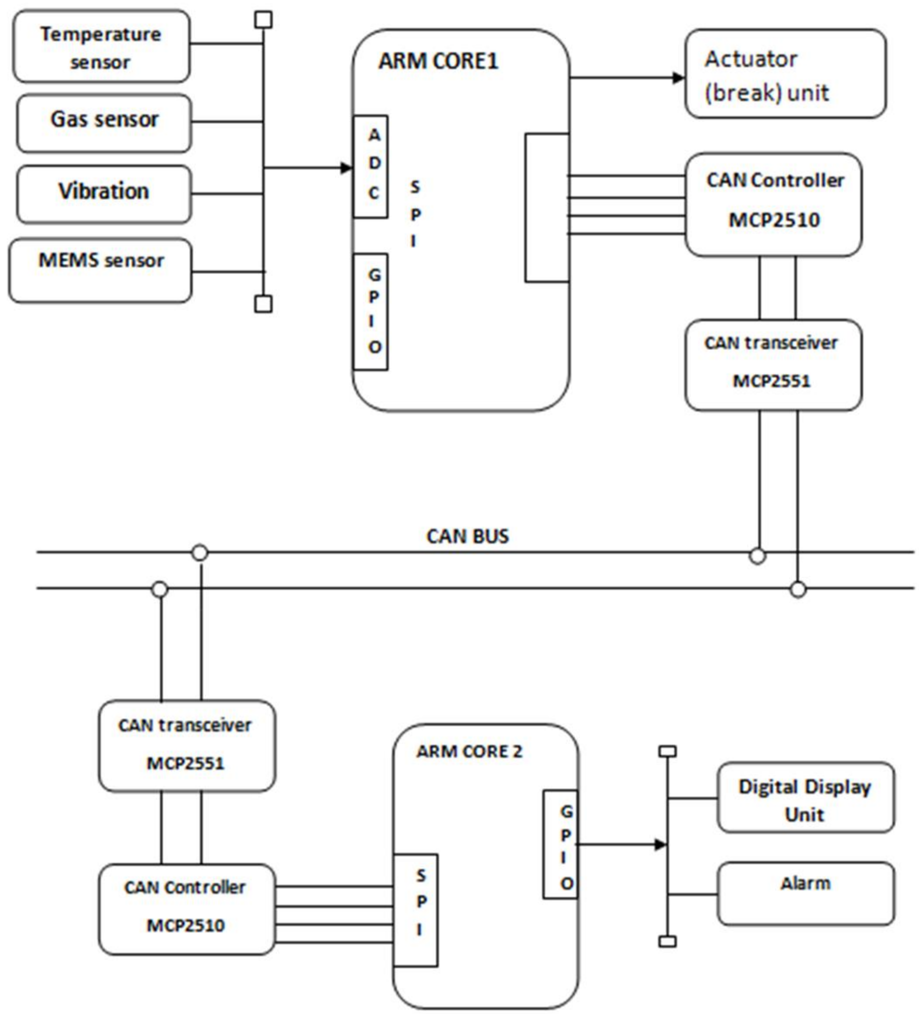
- **Simple & low cost -** ECUs communicate via a single CAN system instead of via direct complex analogue signal lines - reducing errors, weight, wiring and costs.

- **Easy access -** provides 'one point-of-entry' to communicate with all network ECUs - enabling central diagnostics, data logging and configuration.

- **Extremely robust -** robust towards electric disturbances and electromagnetic interference - ideal for safety critical applications (e.g. vehicles).

- **Efficient -** CAN frames are prioritized by ID so that top priority data gets immediate bus access, without causing interruption of other frames or CAN errors.

Controller Node Components



Source: https://www.pantechsolutions.net/a-can-protocol-based-embedded-system-to-avoid-rear-end-collision-of-vehicles

References

- https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial
- https://www.ni.com/en-in/shop/seamlessly-connect-to-third-party-devices-and-supervisory-system/controller-area-network--can--overview.html
- https://www.javatpoint.com/can-protocol