



# MECH5170M

## Connected and Autonomous Vehicles Systems

Communication protocols and Network  
(CAN, Flex-ray, SPI, I<sup>2</sup>C)

Kris Kubiak ( [k.kubiak@leeds.ac.uk](mailto:k.kubiak@leeds.ac.uk) )



2

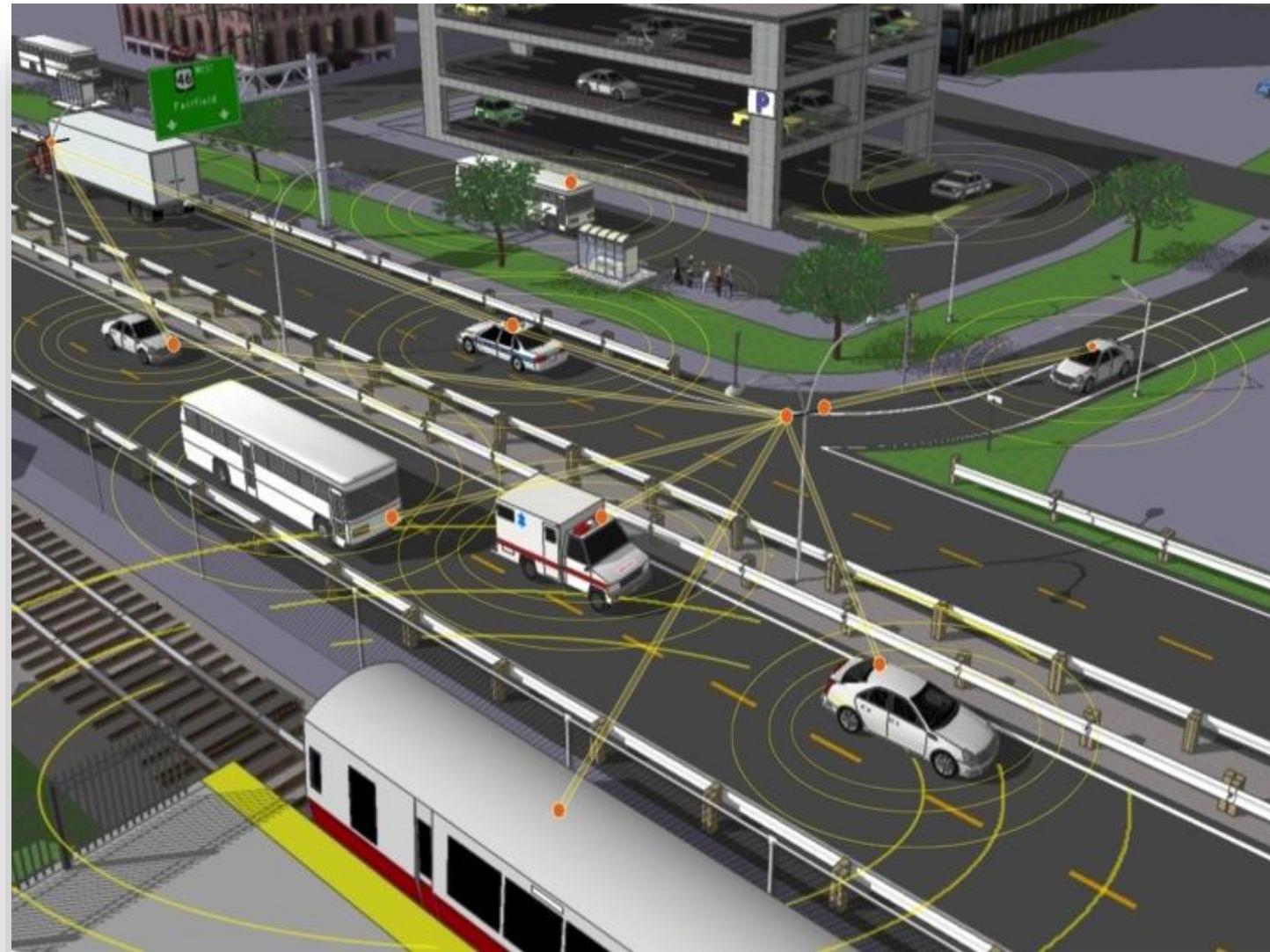
UNIVERSITY OF LEEDS

# Connected Vehicles

# Smar City Concept

## Smart Cities and Connected Vehicles

Smart Cities incorporate and expand connected transportation to ensure that connected transportation data, technologies and applications – as well as connected travellers – are fully integrated with other systems across a city, and fulfil their potential to improve safety, mobility and environmental outcomes.



# Vehicle-to-everything - V2X

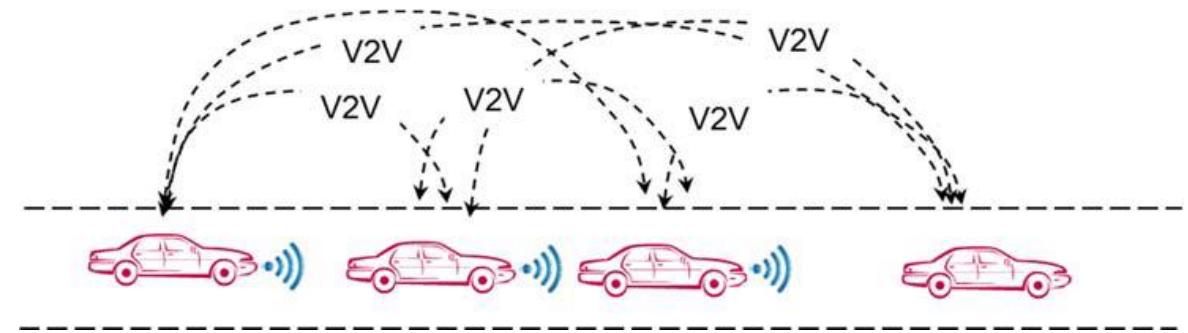
Vehicle-to-everything (V2X) is communication between a vehicle and any entity that may affect, or may be affected by, the vehicle.

It is a vehicle communication system that incorporates other more specific types of communication as:

- V2I (vehicle-to-infrastructure),
- V2N (vehicle-to-network),
- V2V (vehicle-to-vehicle),
- V2P (vehicle-to-pedestrian),
- V2D (vehicle-to-device).

## Vehicle-to-Vehicle (V2V) Communications

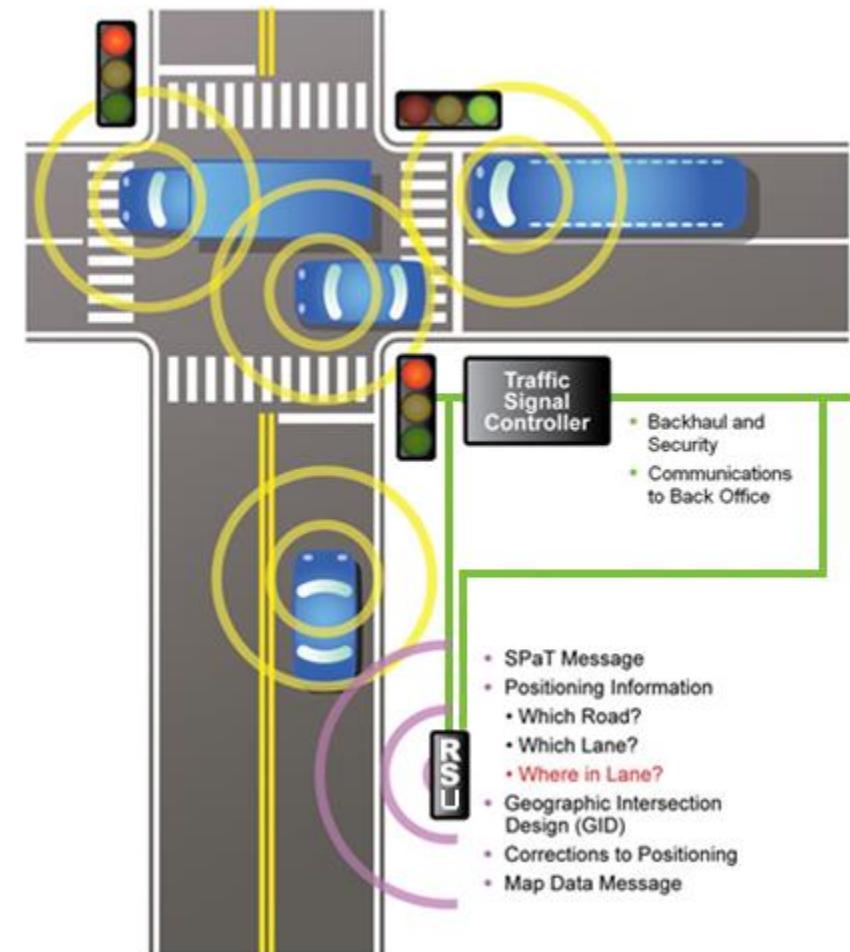
- Allows nearby vehicles to exchange data on their position and use these data to warn drivers of potential collisions
- V2V technologies are capable of warning drivers of potential collisions that are not visible to sensors, such as a stopped vehicle blocked from view, or a moving vehicle at a blind intersection
- Unprecedented and transformative technology: Extendable to other vehicle types, road users, and infrastructure



# Connected Vehicles

## Vehicle-to-Infrastructure (V2I) Communications

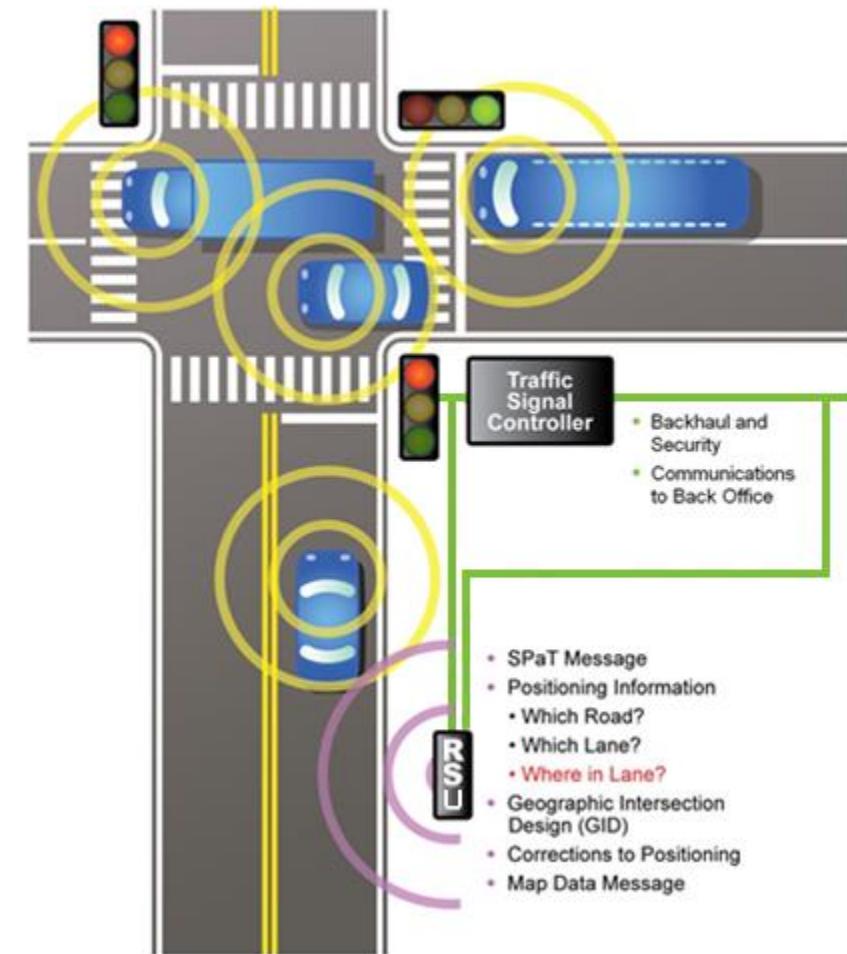
- Allows infrastructure to communicate with vehicles
- Could be used to inform drivers about weather, traffic, work zones, and even potholes
- Allows for coordinated signal timing and enhanced parking information systems that may improve urban traffic flow



# Connected Vehicles

## Vehicle-to-Device (V2D) Communications

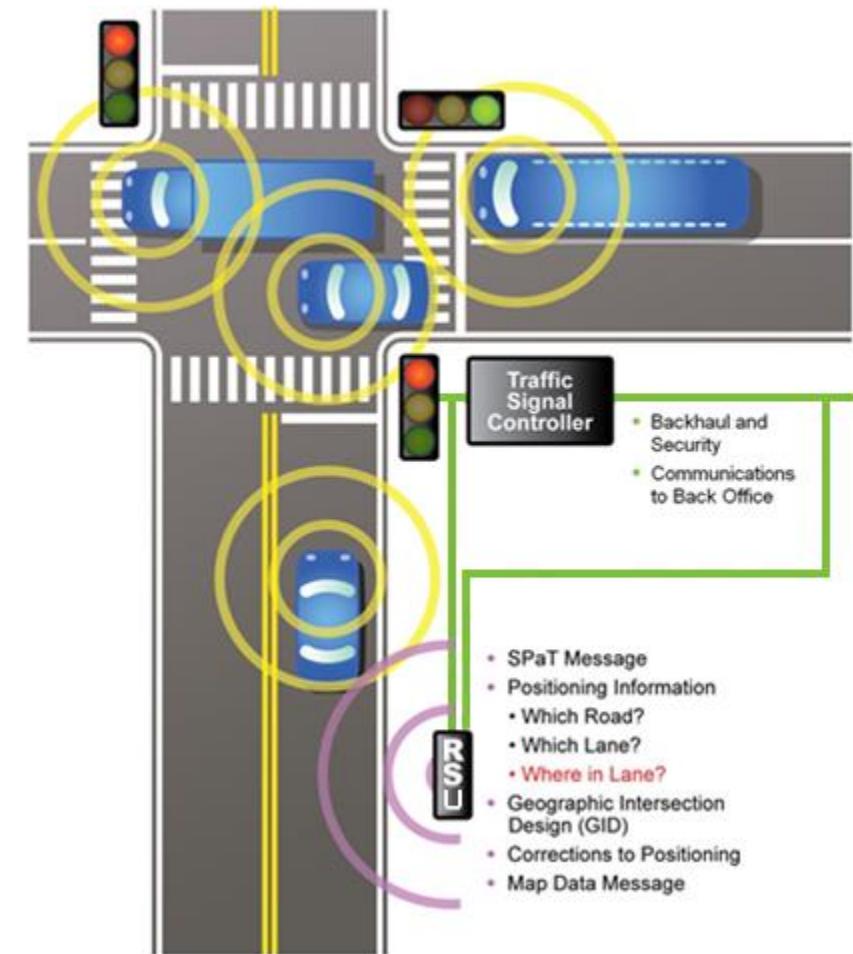
- Vehicle-to-device (V2D) communication is the exchange of information between a vehicle and any electronic device that may be connected to the vehicle itself.
- Vehicle connectivity with mobile apps have the great potential to offer a better driving experience, by providing information regarding the surrounding vehicles and infrastructure. For example, keyless car will communicate with the owner's phone.
- Pedestrian phone could warn the vehicle about its position or potential collision.



# Connected Vehicles

## Vehicle-to-Network (V2N) Communications

- Vehicle to Network aims to transmit information between vehicles and the management system.
- This process is made possible through high-bandwidth, low-latency, high-reliability network infrastructure.
- Cars can receive broadcast alerts about traffic congestion or accidents further down the road to pave the way for autonomous driving with the future of mobility.



**Digital bandwidth:** The rate at which data can be transferred

**Latency:** The delays inherent in moving data from source to destination

**Reliability:** The likelihood that a message will be received at its destination

**Security and authentication:** The ability to verify the identity of the source of a message and to guarantee that its contents has not been tampered with or read by an unauthorised receiver

**Network configuration:** For example, broadcast, broadcast with forwarding, or point-to-point routed communication topologies

# Connected Vehicles, What type of data can be received

UNIVERSITY OF LEEDS

V2I Safety	Environment	Mobility
Red Light Violation Warning Curve Speed Warning Stop Sign Gap Assist Spot Weather Impact Warning Reduced Speed/Work Zone Warning Pedestrian in Signalized Crosswalk Warning (Transit)	Eco-Approach and Departure at Signalized Intersections Eco-Traffic Signal Timing Eco-Traffic Signal Priority Connected Eco-Driving Wireless Inductive/Resonance Charging Eco-Lanes Management Eco-Speed Harmonization Eco-Cooperative Adaptive Cruise Control Eco-Traveler Information Eco-Ramp Metering Low Emissions Zone Management AFV Charging / Fueling Information Eco-Smart Parking Dynamic Eco-Routing (light vehicle, transit, freight) Eco-ICM Decision Support System	Advanced Traveler Information System Intelligent Traffic Signal System (I-SIG) Signal Priority (transit, freight) Mobile Accessible Pedestrian Signal System (PED-SIG) Emergency Vehicle Preemption (PREEMPT) Dynamic Speed Harmonization (SPD-HARM) Queue Warning (Q-WARN) Cooperative Adaptive Cruise Control (CACC) Incident Scene Pre-Arrival Staging Guidance for Emergency Responders (RESP-STG) Incident Scene Work Zone Alerts for Drivers and Workers (INC-ZONE) Emergency Communications and Evacuation (EVAC) Connection Protection (T-CONNECT) Dynamic Transit Operations (T-DISP) Dynamic Ridesharing (D-RIDE) Freight-Specific Dynamic Travel Planning and Performance Drayage Optimization
V2V Safety	Road Weather	Smart Roadside
Emergency Electronic Brake Lights (EEBL) Forward Collision Warning (FCW) Intersection Movement Assist (IMA) Left Turn Assist (LTA) Blind Spot/Lane Change Warning (BSW/LCW) Do Not Pass Warning (DNPW) Vehicle Turning Right in Front of Bus Warning (Transit)	Motorist Advisories and Warnings (MAW) Enhanced MDSS Vehicle Data Translator (VDT) Weather Response Traffic Information (WxTINFO)	Wireless Inspection Smart Truck Parking
Agency Data		
Probe-based Pavement Maintenance Probe-enabled Traffic Monitoring Vehicle Classification-based Traffic Studies CV-enabled Turning Movement & Intersection Analysis CV-enabled Origin-Destination Studies Work Zone Traveler Information		

# CAV Hardware - Communication Protocols

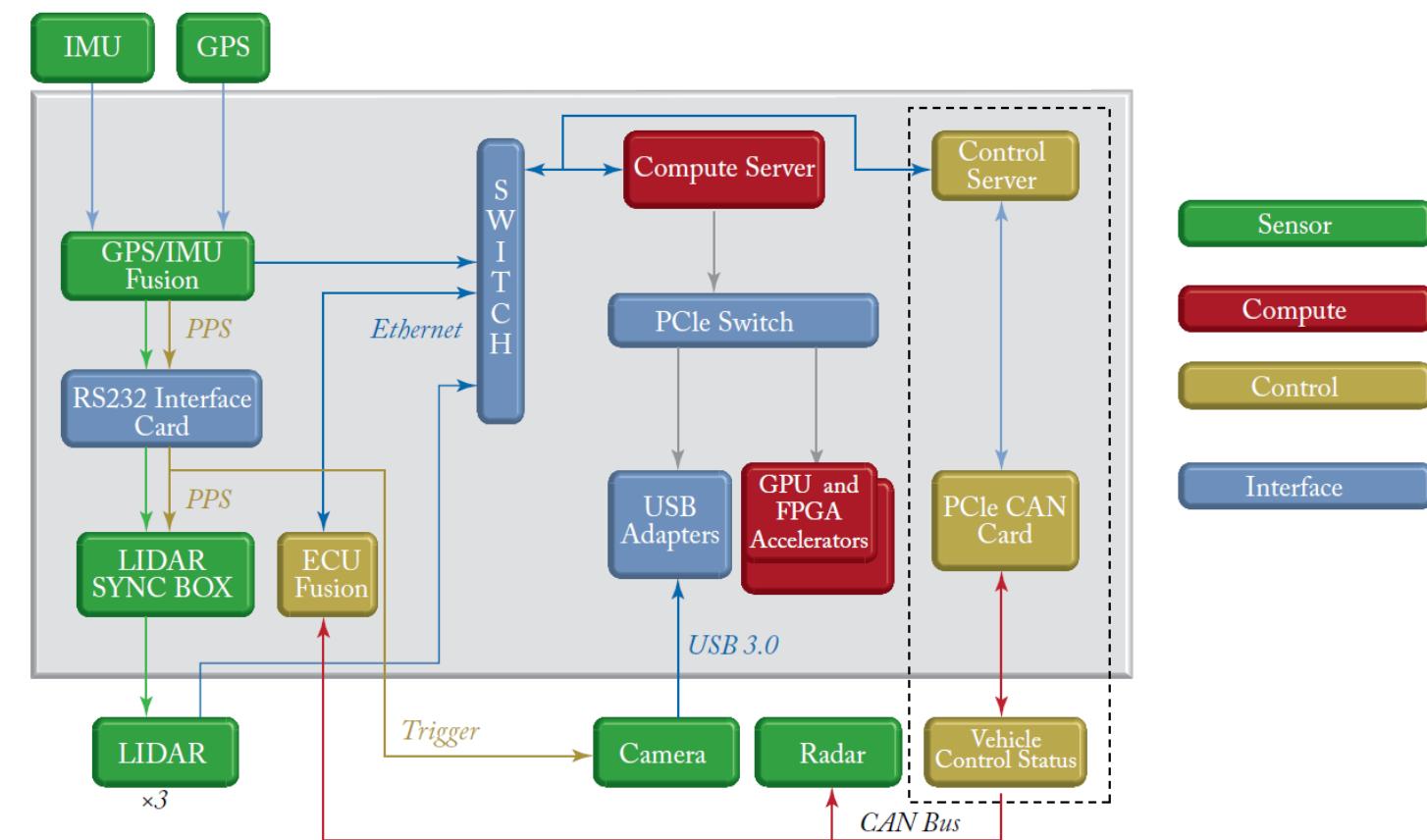
**USB** - Universal Serial Bus

**CAN Bus** - Connected Area Network

**Ethernet LAN** - local Area Network

**SPI** - Serial Peripheral Interface

**I<sup>2</sup>C** - Inter-Integrated Circuit



# Very Short Range SPI, I<sub>2</sub>C

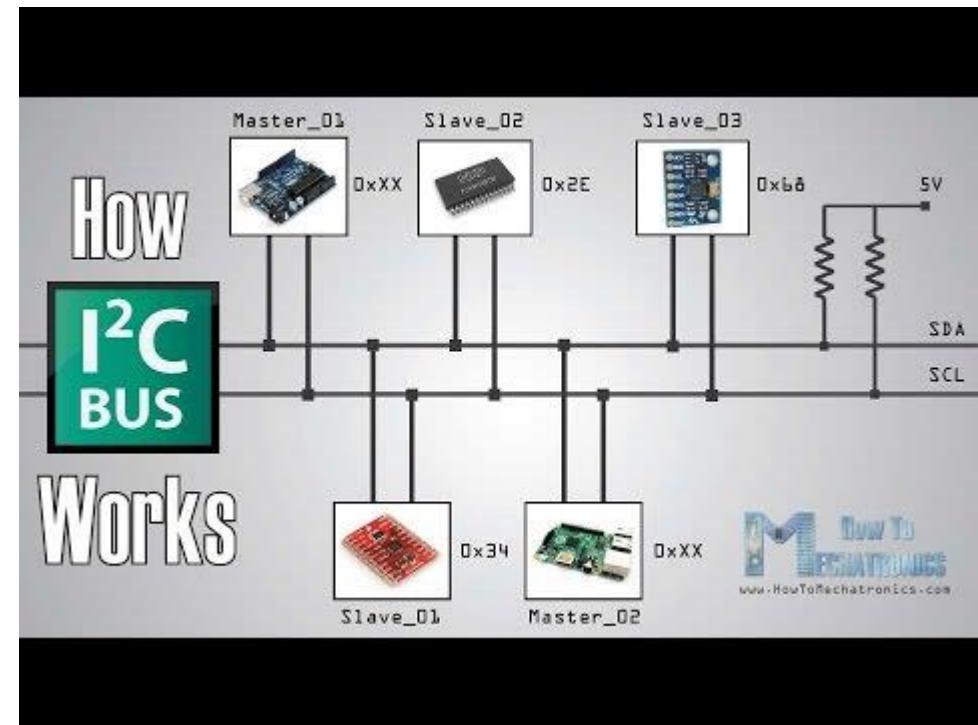
# I<sup>2</sup>C - Inter-Integrated Circuit

## I<sup>2</sup>C - Inter-Integrated Circuit

It is a synchronous, multi-master/multi-slave, packet switched, single-ended, serial communication bus invented in 1982 by Philips Semiconductors.

It is widely used for attaching lower-speed peripheral ICs to processors and microcontrollers in short-distance, intra-board communication.

The **speed** grades (standard mode: **100 kbit/s**, full speed: **400 kbit/s**, fast mode: **1 mbit/s**, high speed: **3,2 Mbit/s**) are maximum ratings.



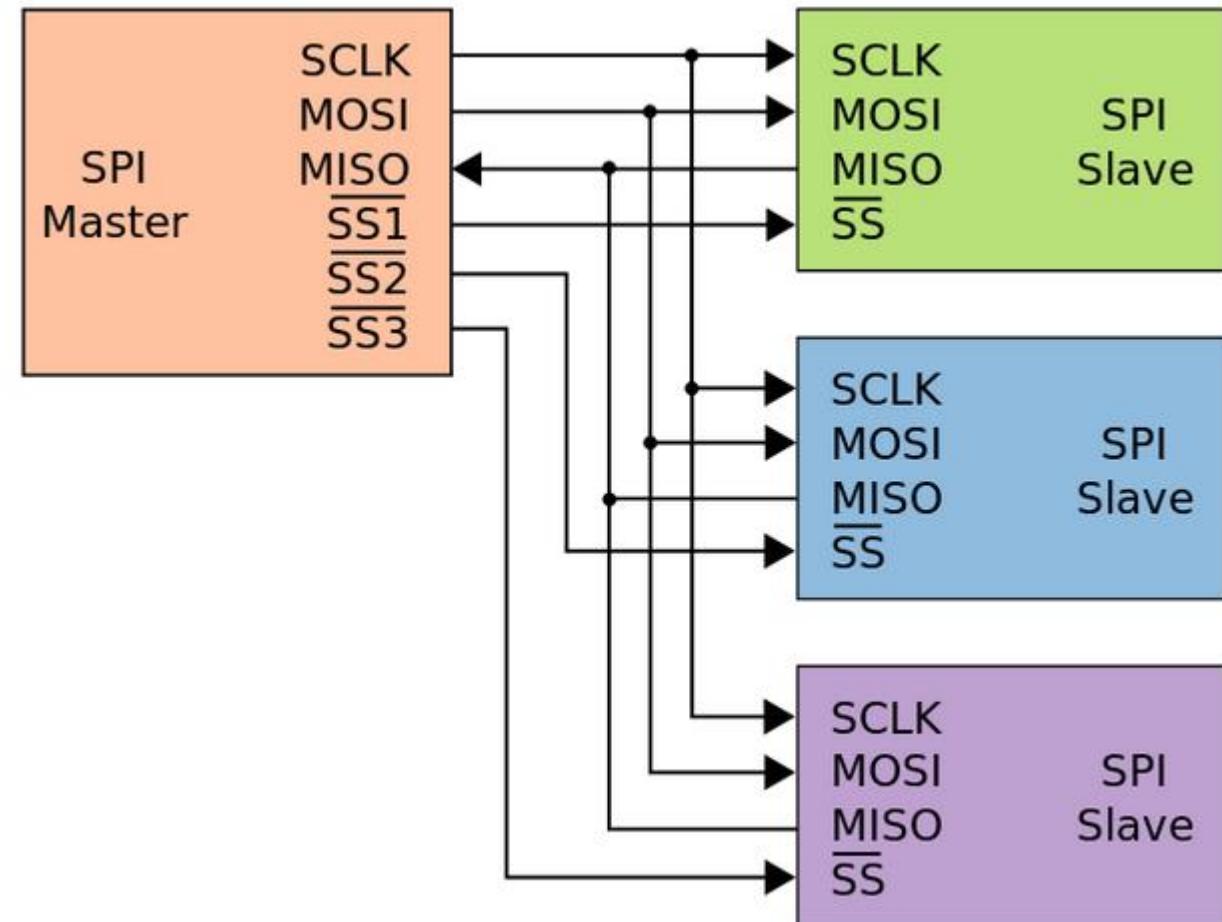
# SPI - Serial Peripheral Interface

SPI (serial peripheral interface) busses are a favourite of designers for many reasons.

The SPI bus can run at high speed, transferring data at **up to 60 Mbps** over short distances like between chips on a board.

The bus is conceptually simple, consisting of a:

- Clock (**SCLK**)
- Two data lines (**MOSI**, **MISO**)
- Chip select signal (**SS**)





15

UNIVERSITY OF LEEDS

# CAN Bus

# Controller Area Network (CAN)

Controller Area Network (CAN) is a fast serial bus that is designed to provide

- an efficient,
- Reliable and
- Very economical link between sensors and actuators.

CAN uses a twisted pair cable to communicate at speeds up to **1Mbit/s** with up to **40 devices**.

Originally developed to simplify the wiring in automobiles.

CAN fieldbuses are now used in machine and factory automation products as well.



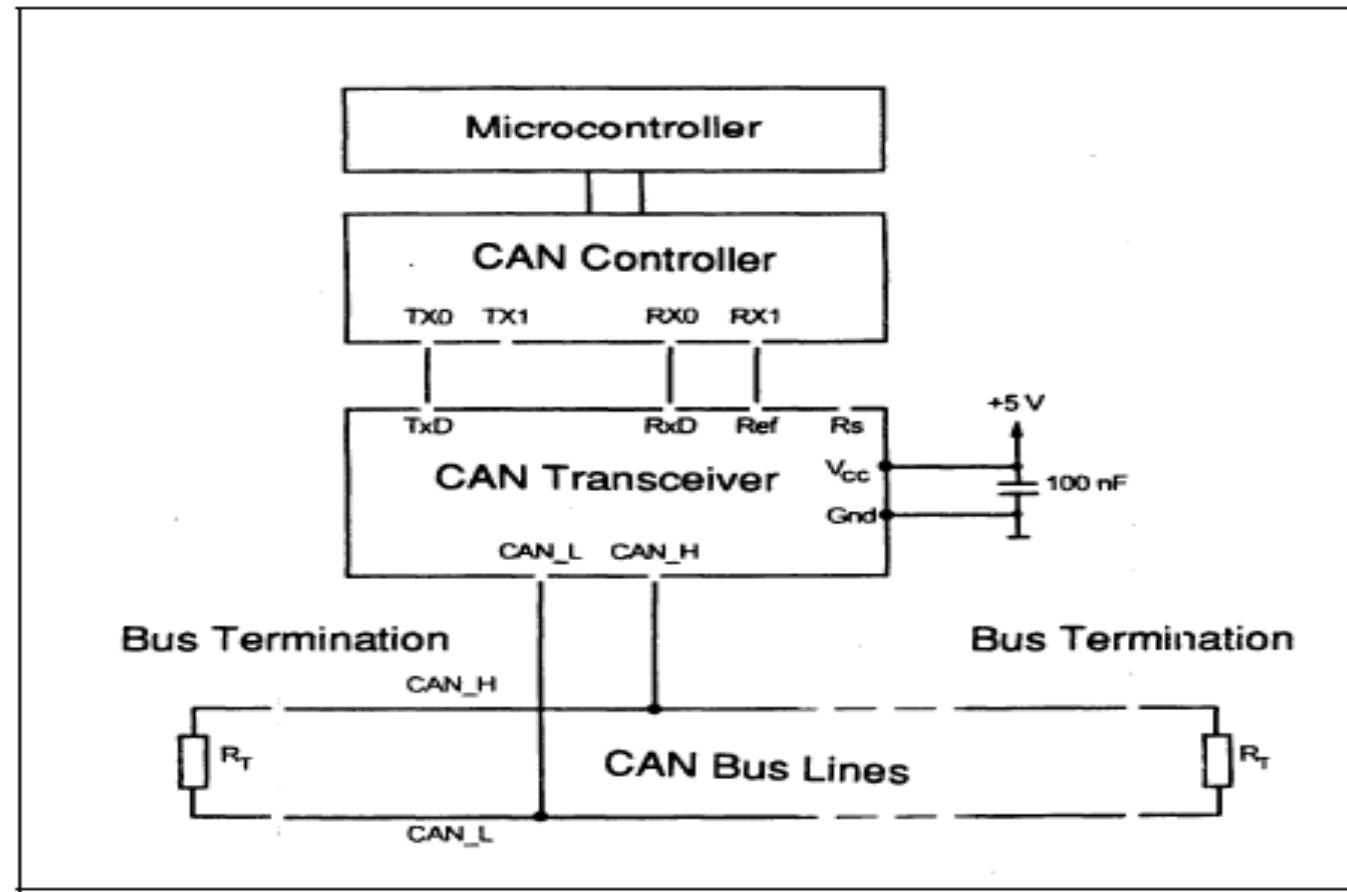
# CAN features

- Any node can access the bus when the bus is quiet
- Non-destructive bit-wise arbitration to allow 100% use of the bandwidth without loss of data
- Variable message priority based on 11-bit (or 29 bit) packet identifier
- Peer-to-peer and multi-cast reception
- Automatic error detection, signaling and retries
- Data packets are 8 bytes long



- By introducing one single bus as the only means of communication as opposed to the point-to-point network, we traded off the channel access simplicity for the circuit simplicity
- Since two devices might want to transmit simultaneously, we need to have a MAC protocol to handle the situation.
- CAN manages MAC issues by using a **unique identifier** for each of the outgoing messages
- Identifier of a message represents its **priority**.

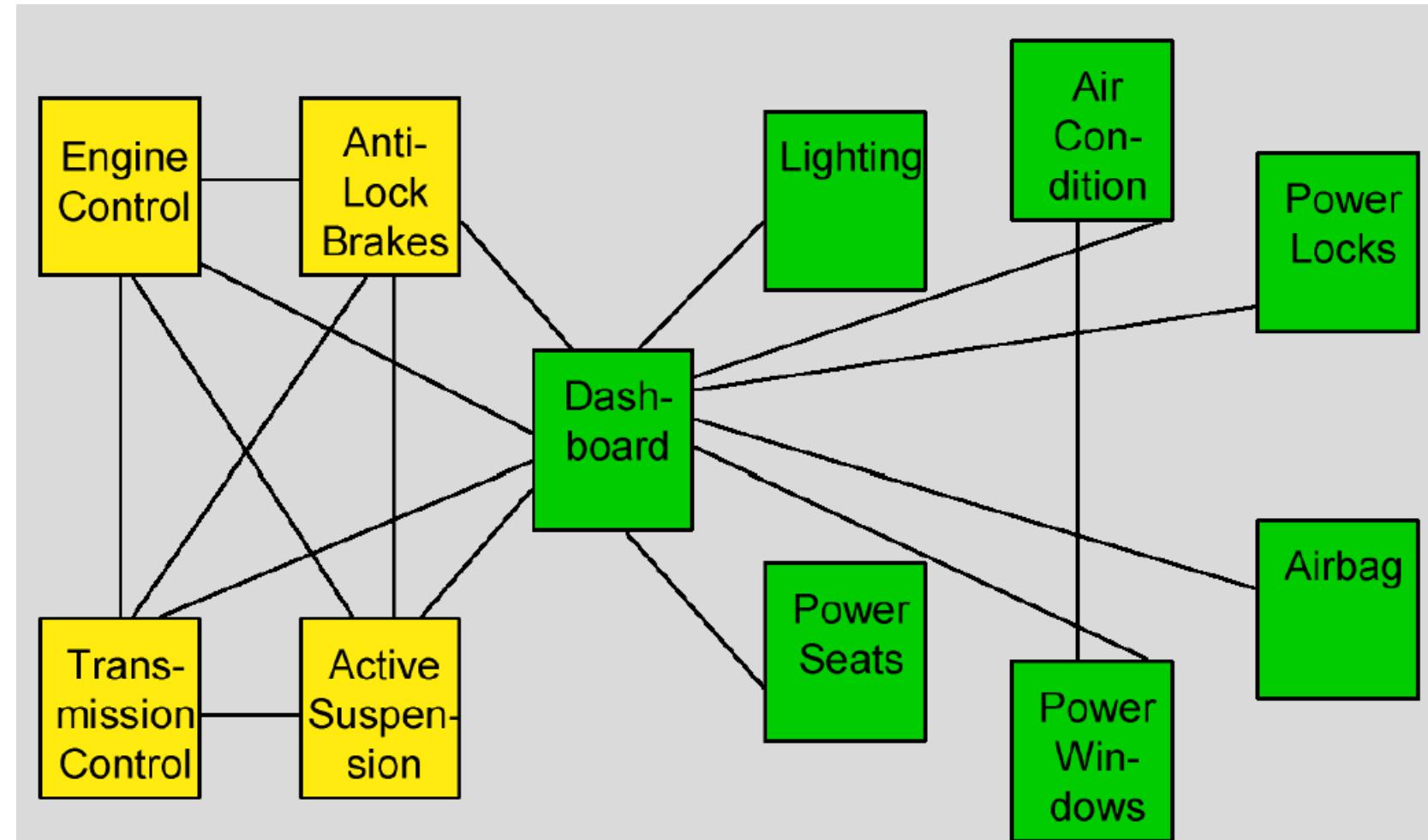
# Physical CAN connection



Physical CAN Connection according to ISO 11898

# Before CAN

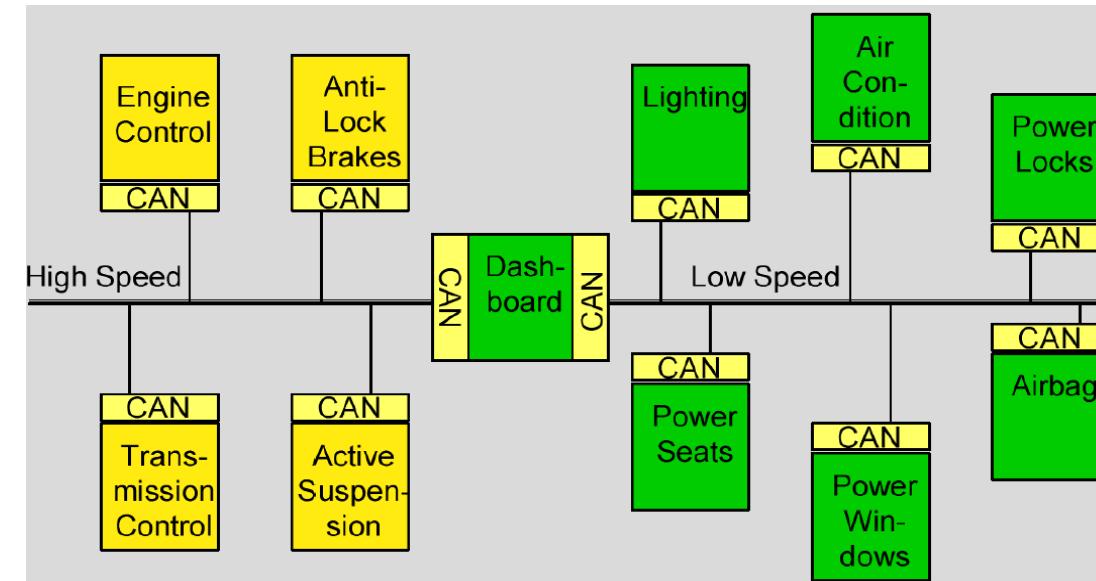
## Point-to-point wiring



# With CAN

The solution to this problem was the connection of the control systems via a serial bus system. With the use of CAN, point-to-point wiring is replaced by one serial bus connecting all control systems.

This is accomplished by adding some CAN-specific hardware to each control unit that provides the "rules" or the protocol for transmitting and receiving information via the bus.



## Message arbitration - priority

Each ECU needs to have unique **identifier** that is also **representing its priority**.

For example ECU1 with identifier 0x09 with have a priority above the ECU2 with identified 0x0A.

Hex value of 0x09 is 9 in decimal and 0x0A is 10 ... 0x0F is 15, 0xFF is 255 etc...

Arbitration compares bits starting from the most significant bit. ECU1 has **lower identifier (0x09)**, meaning it has the **higher priority**. It will dominate the bus during arbitration and continue transmitting, while ECU2 will have to wait.



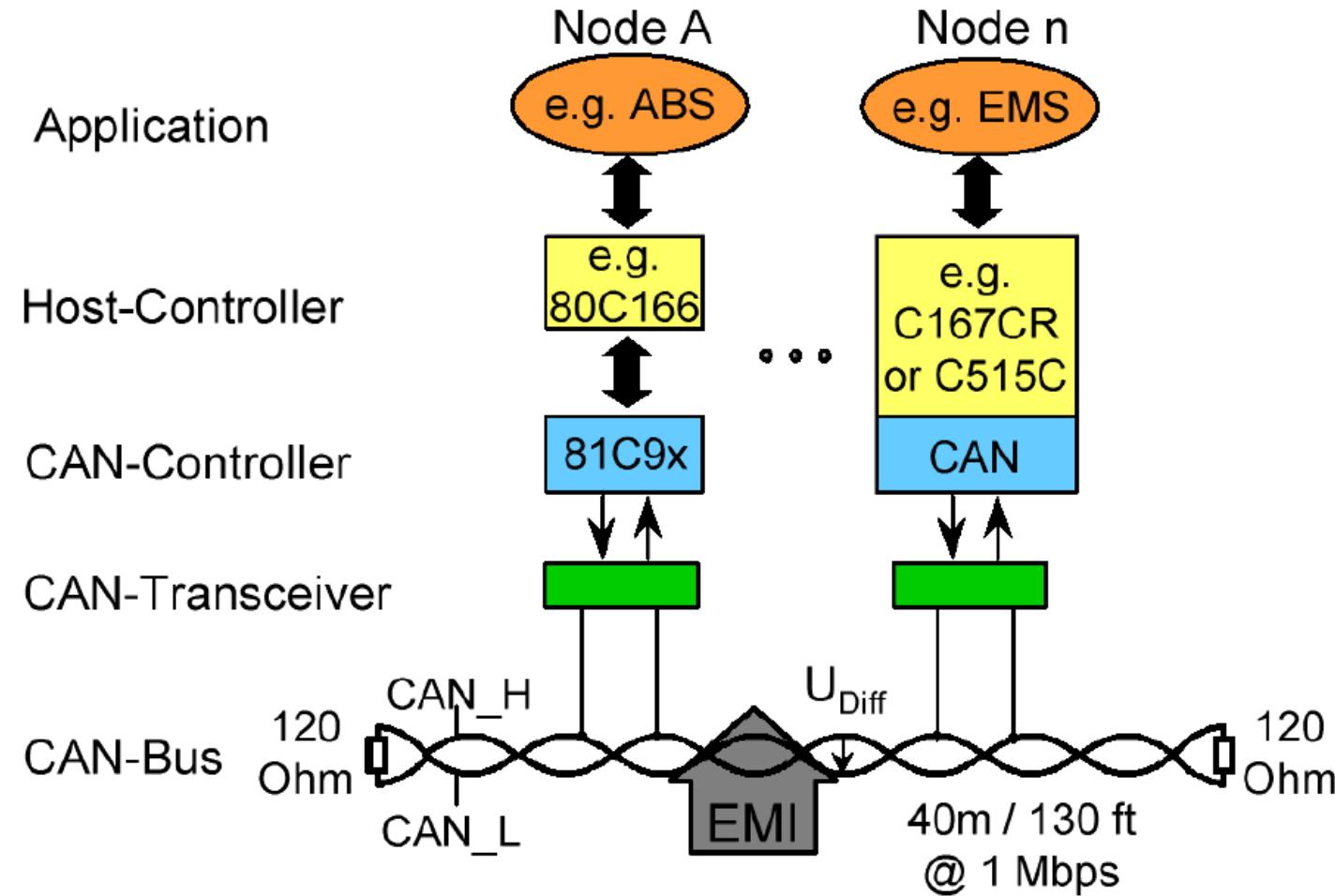
## Object, Transfer, and Physical Layers:

- **Object Layer:** handles messages - selects transmit/receive messages
- **Transfer Layer:** assures messages adheres to protocol
- **Physical Layer:** sends and receives messages

# Physical Layer

- Topology
  - Terminated bus
- Number of stations
  - In principle limited to 30 (depends on drivers)
- Medium
  - Twisted pair, single wire
- Range
  - Signaling speed and propagation speed dependent: 40m at 1Mbit/s
- Signaling and bit encoding
  - 10 kbit/s to 1 Mbit/s, NRZ

# Basic Configuration



# Basic message frame format

Field name	Length (bits)	Purpose
Start-of-frame	1	Denotes the start of frame transmission
Identifier	11	A (unique) identifier for the data
Remote transmission request (RTR)	1	Must be dominant (0)
Identifier extension bit (IDE)	1	Must be dominant (0)
Reserved bit (r0)	1	Reserved bit (it must be set to dominant (0), but accepted as either dominant or recessive)
Data length code (DLC)	4	Number of bytes of data (0-8 bytes)
Data field	0-8 bytes	Data to be transmitted (length dictated by DLC field)
CRC	15	Cyclic redundancy check
CRC delimiter	1	Must be recessive (1)
ACK slot	1	Transmitter sends recessive (1) and any receiver can assert a dominant (0)
ACK delimiter	1	Must be recessive (1)
End-of-frame (EOF)	7	Must be recessive (1)

# CAN Bus Overview

The physical layer uses differential transmission on a twisted pair wire.

The bus uses Non-Return To Zero (NRZ) with bit-stuffing.

The nodes are connected to the bus in a *wired-and* fashion: if just one node is driving the bus to a logical 0, then the whole bus is in that state regardless of the number of nodes transmitting a logical 1.

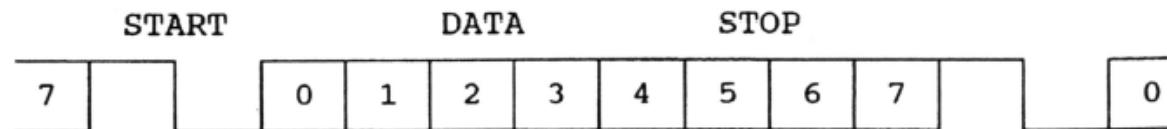
Message length is short with a maximum of 8 data bytes per message and there is a low latency between transmission request and start of transmission.

The messages are protected by a CRC type checksum

# Basic Bit Encoding

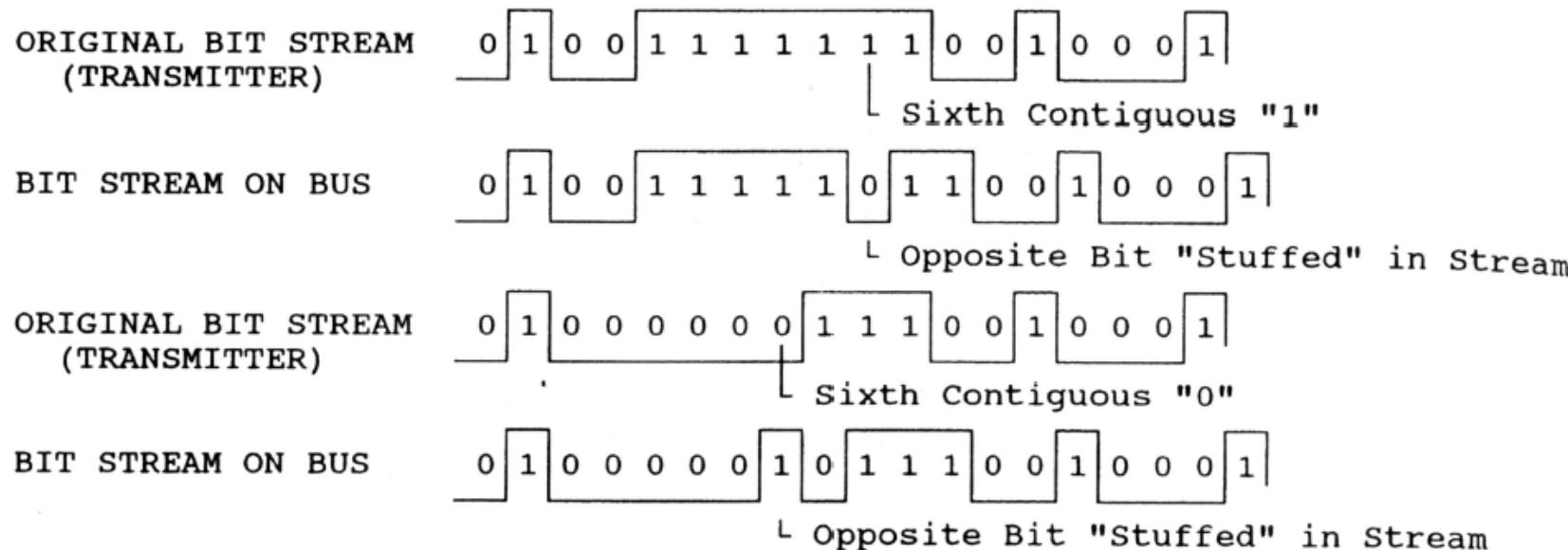
**NRZ = Non-Return-To Zero**

- Fewer transitions (on average) = less EMI, but requires less oscillator drift



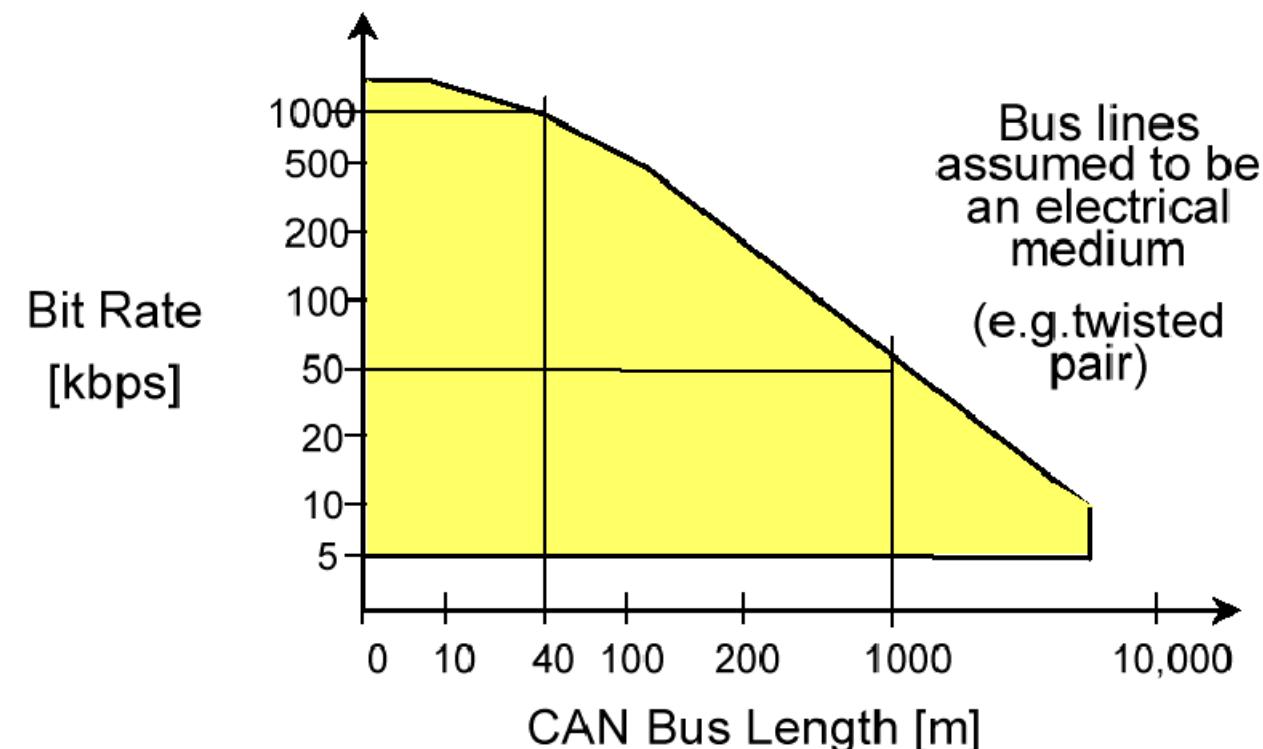
**FIGURE 26.21** A 10-bit NRZ waveform (LSB first)

- Bit stuffing relaxes oscillator drift requirements



# Bus Transmission Speed

Arbitration limits bus speed. Maximum speed =  $2 \times t_{pd}$   
 $t_{pd}$  = propagation delay of electrical medium



## The Can Protocol - Limitations

Specifies how small packets of data may be transported from point A to point B using a shared communications medium.

It (quite naturally) contains nothing on topics such as

- flow control
- transportation of data larger than can fit in a 8-byte message
- node addresses
- establishment of communication, etc.

# Software - Higher layer protocols

## Some high layer protocols (Libraries)

- **CANopen**
- Device net
- CANKingdom

## Higher layer protocols are used in order to

- standardise startup procedures including bit rate setting
- distribute addresses among participating nodes or kinds of messages
- determine the layout of the messages
- provide routines for error handling at the system level

# The CAN Standard

The CAN standard defines four message types

- Data Frame – the predominantly used message type
- Remote Frame
- Error Frame
- Overload Frame

The messages uses a clever scheme of bit-wise arbitration to control access to the bus, and each message is tagged with a priority.

The CAN standard also defines an elaborate scheme for error handling and confinement.

CAN may implemented using different physical layers, and there are also a number of different connector types in use.

# 1. The Data Frame

**Summary: "Hello everyone, here's some data labeled X, hope you like it!"**

**The Data Frame is the most common message type.**

It comprises the following major parts (a few details are omitted for the sake of brevity):

the **Arbitration Field**, which determines the priority of the message when two or more nodes are contending for the bus. The Arbitration Field contains:

- For CAN 2.0A, an 11-bit Identifier and one bit, the RTR bit, which is dominant for data frames.
- For CAN 2.0B, a 29-bit Identifier (which also contains two recessive bits: SRR and IDE) and the RTR bit.

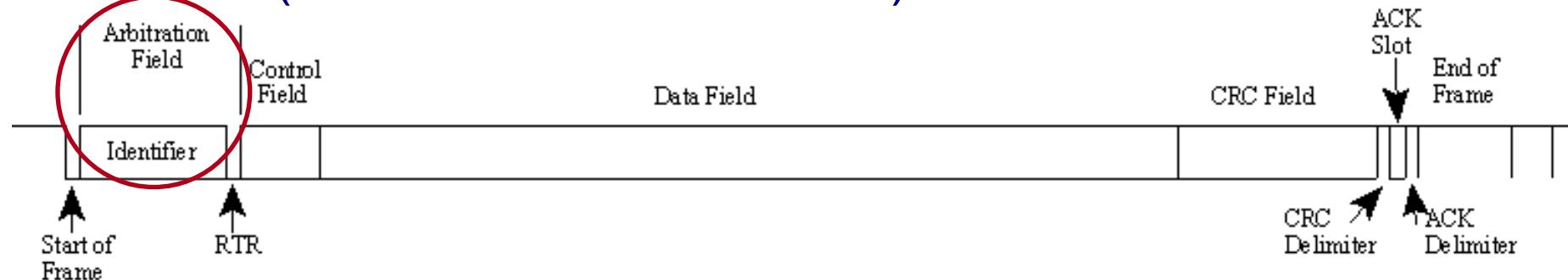
the **Data Field**, which contains zero to eight bytes of data.

the **CRC Field**, which contains a 15-bit checksum calculated on most parts of the message. This checksum is used for error detection.

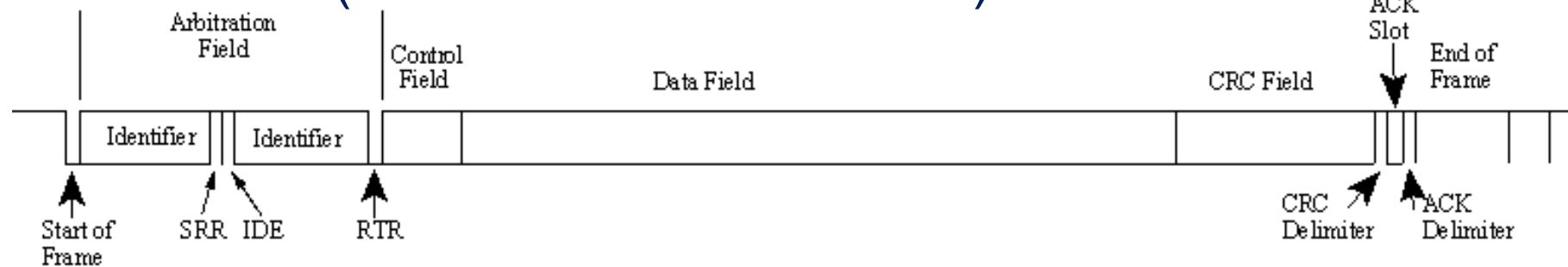
an **Acknowledgement Slot**; **any** CAN controller that has been able to correctly receive the message sends an Acknowledgement bit at the end of each message. The transmitter checks for the presence of the Acknowledge bit and retransmits the message if no acknowledge was detected.

# CAN Data Frames

## CAN 2.0A (“standard CAN” 11-bit ID) Data Frame.



## ► CAN 2.0B (“extended CAN” 29-bit ID) Data Frame.



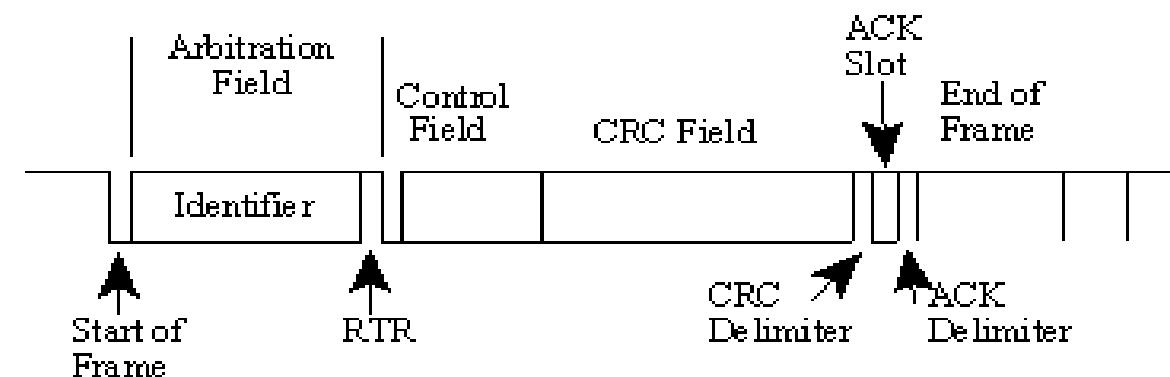
Note 1: It is worth noting that the presence of an Acknowledgement Bit on the bus does not mean that any of the *intended* addressees has received the message. The only thing we know is that *one or more* nodes on the bus has received it correctly

## 2. Remote Frame

There's one catch with the Remote Frame: the Data Length Code *must be set to the length of the expected response message*. Otherwise the arbitration will not work.

Sometimes it is claimed that the node responding to the Remote Frame is starting its transmission as soon as the identifier is recognized, thereby "filling up" the empty Remote Frame. ***This is not the case.***

A Remote Frame (2.0A type):

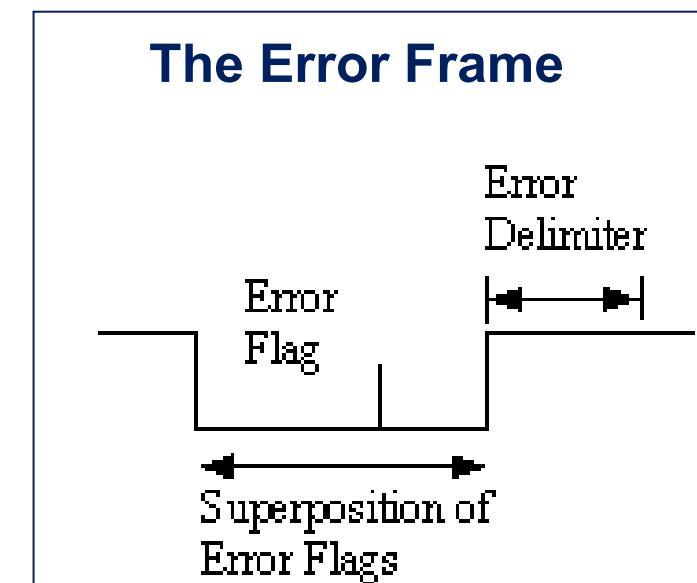


### 3. The Error Frame

#### ***Summary: (everyone, aloud) "OH DEAR, LET'S TRY AGAIN"***

Simply put, the Error Frame is a special message that violates the framing rules of a CAN message. It is transmitted when a node detects a fault and will cause all other nodes to detect a fault - so they will send Error Frames, too. The transmitter will then automatically try to retransmit the message. There is an elaborate scheme of error counters that ensures that a node can't destroy the bus traffic by repeatedly transmitting Error Frames.

The Error Frame consists of an Error Flag, which is 6 bits of the same value (thus violating the bit-stuffing rule) and an Error Delimiter, which is 8 recessive bits. The Error Delimiter provides some space in which the other nodes on the bus can send their Error Flags when they detect the first Error Flag.





## 4. The Overload Frame

***Summary: "I'm a very busy little 82526 device, could you please wait for a moment?"***

The Overload Frame is mentioned here just for completeness. It is very similar to the Error Frame with regard to the format and it is transmitted by a node that becomes too busy. The Overload Frame is not used very often, as today's CAN controllers are clever enough not to use it. In fact, the only controller that will generate Overload Frames is the now obsolete 82526



# FlexRay

# FlexRay

The FlexRay bus is a deterministic, fault-tolerant, and high-speed bus system developed in conjunction with automobile manufacturers.

## 1. General

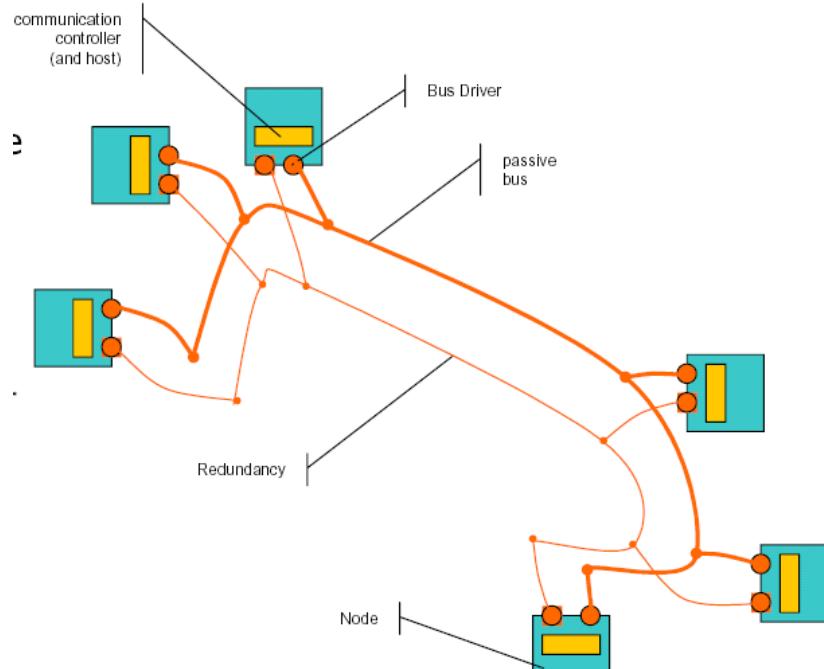
- Higher bandwidth
- Fault tolerance
- Deterministic data transmission with guaranteed latency and minimal jitter.
- Support for distributed systems

## 2. Automotive

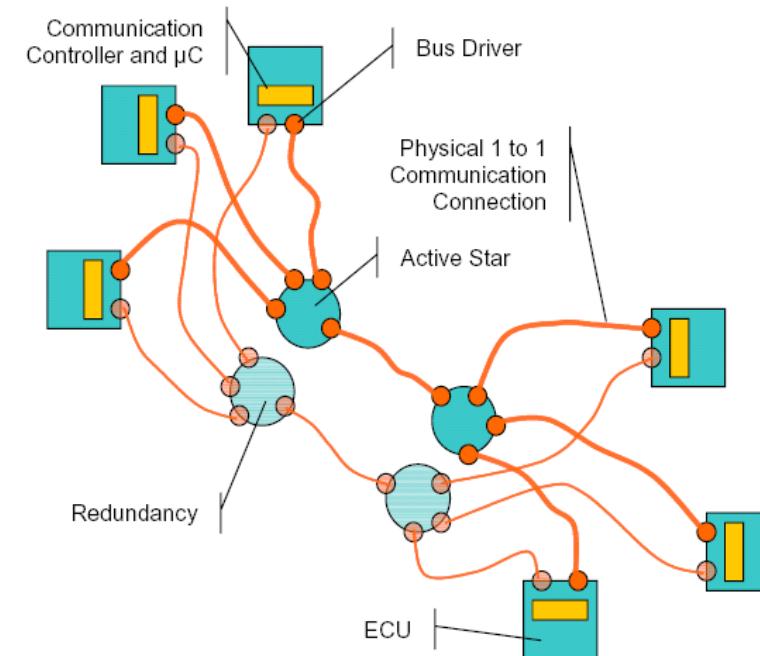
- Configurable synchronous and asynchronous transmission
- Prompt error detection and error reporting.
- Fault-containment at the level of the physical layer.
- Support for a fiber-optics and electrical physical layer.
- Flexibility, expandability and easy configuration in automotive applications.

# Topology

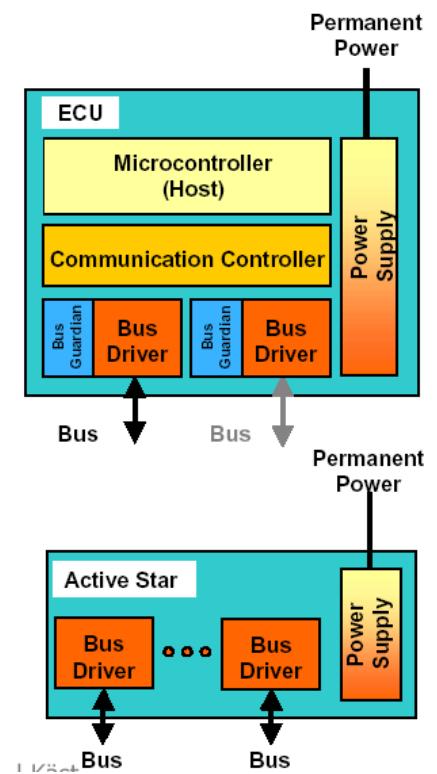
**BUS**



**STAR**



**Hybrid**

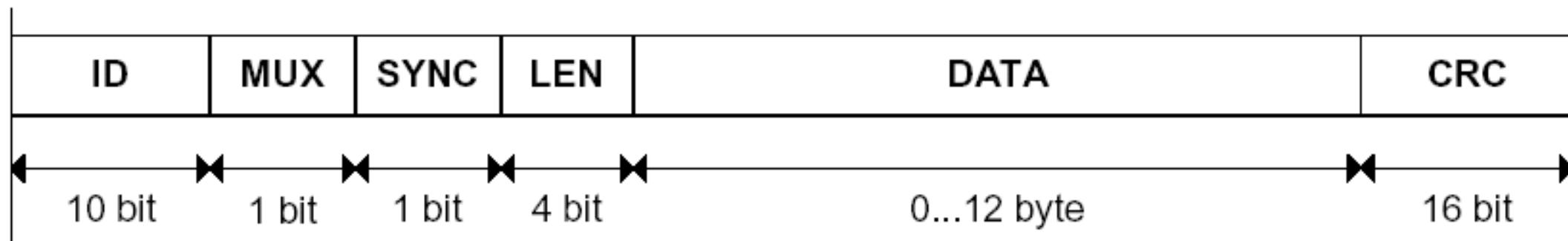


# FlexRay - Data Frame

Max data field length = 12B

Schedule determined at runtime

Have 2 channels



# CAN and FlexRay Comparison

By designing network configurations ahead of time, network designers save significant cost and increase reliability of the network.

For a TDMA network such as FlexRay to work correctly, all nodes must be configured correctly.

The FlexRay standard is adaptable to many different types of networks and tailor update speeds, data volume.

	CAN	FlexRay
Bandwidth	1 Mbps	10 Mbps
Number of channels	1	2
Frame data length	0~8	0~254
Communication	Dynamic arbitration	TDMA
Complexity	Low	High
Composability	No	Yes
Flexibility	One topology	Many different topologies



43

UNIVERSITY OF LEEDS

USB

# USB Speed

USB Versions			Sign	Transfer Rate	Theoretical Speed
Current Version Name	Market Code	Original Name			
USB 2.0	LowSpeed	Low Speed	USB 1.0		1.5Mbps 0.1875MB/s
	FullSpeed	Full Speed	USB 1.1		12Mbps 1.5MB/s
	HiSpeed	Hi-Speed	USB 2.0		480Mbps 60MB/s
USB 3.2	Gen 1	SuperSpeed USB	USB 3.1GEN1		5Gbps 625MB/s
	Gen 2	SuperSpeed USB 10Gbps	USB 3.1GEN2		10Gbps 1250 MB/s
	Gen 2x2	SuperSpeed USB 20Gbps	N/A		20Gbps 2500 MB/s
USB 4				40Gbps	5GB/s



45

UNIVERSITY OF LEEDS

# Mobile Data

# Mobile Data Connectivity and 5G

**Latency** - The real-time operational data communications in smart grid include online sensor reading and power system control signals.

For real-time sensing purposes, reading messages should be transmitted within a very short time frame.

For instance, the maximum allowed time is in the range of 12–20 ms.

**Bandwidth** - Autonomous vehicles are expected to generate about 40TB of data from sensors and download data as they are connected to each other and other equipment on the road. One of the key technologies that may provide an adequate platform to transmit this enormous amount of data with a transmission speed that enables self-driving cars to make effective decisions in real-time road conditions is **5G** technology.

# GSO Satellite Connection

**High Latency of GSO Satellite Connection** (when mobile network not available)

Because they are very far from Earth, over 22,000 miles away, GSO satellites have at least 250ms latency.

By comparison, mobile network signal latency is about 10ms.

Satellite connectivity can be **used as a last resort**, it is **not suitable** for normal operation of autonomous vehicles.

## 5G latency

Third-generation mobile networks 3G had latencies in the **hundreds of milliseconds**.

4G networks started with latencies of about 100ms and now are down to a range of about **30ms to 70ms**. That's getting closer to the theoretical 4G latency of just 10ms.

With 5G networks, with good networks somewhere between 5ms and 20ms. But that's just today's latency. The ultimate goal for 5G, set by an industry group called the 3GPP (3rd Generation Partnership Project) hopes 5G network improvements ultimately can push latency all the way **down to 1ms**.

# Conclusions

- Short range protocols - SPI, I2C
- Mid range communication CAN, FleXRay
- Slow speed vs high speed USB
- High speed, high bandwidth, WLAN, 4G, 5G

ANY QUESTIONS  
???