Automotive Connectivity
Module 1: Lesson 4

Carlo Augusto Grazia

Tenure-Track Assistant Professor

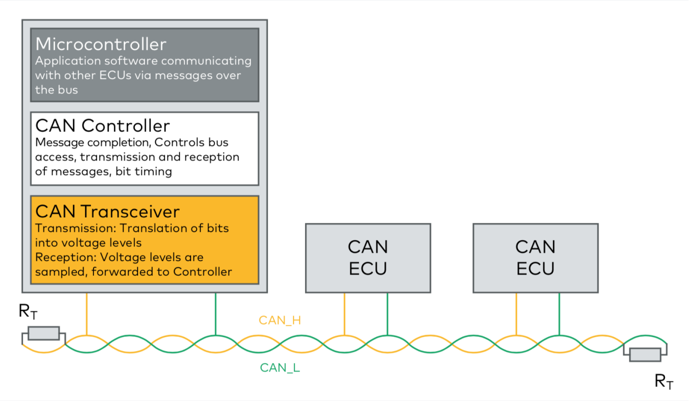
Department of Engineering *Enzo Ferrari*
University of Modena and Reggio Emilia

Modena, 24th September 2024

The **DARK** electronic side (2) of the CAN Bus

Going to the bottom of the stack

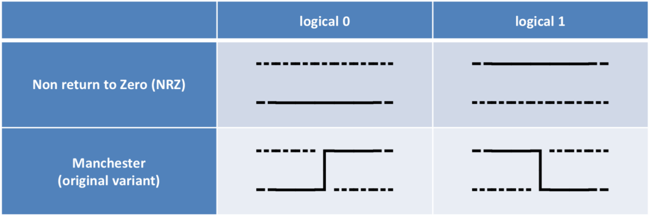
Do you remember this figure?

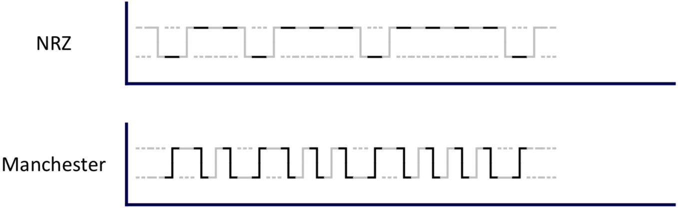


We need to deal with the physical layer

Bit Coding

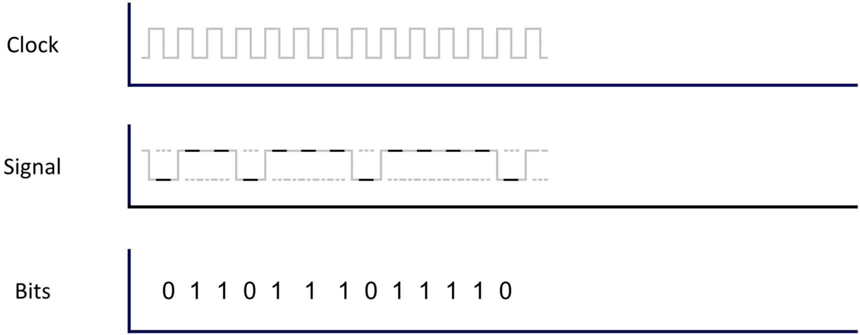
How to code bits? Two main coding techniques:





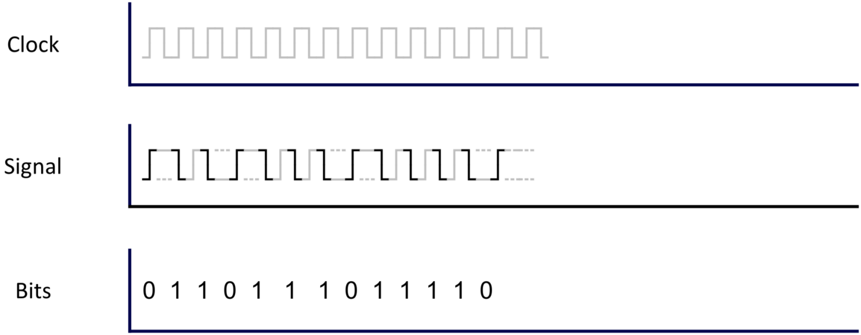
NRZ Coding

A string of bit in NRZ coding



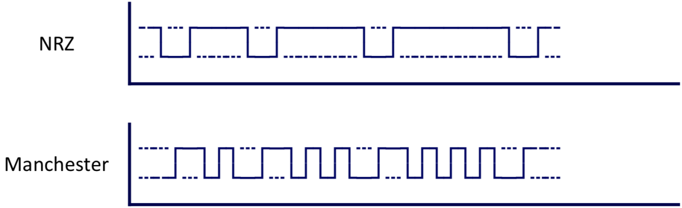
Manchester Coding

A string of bit in Manchester coding



Bit Coding: Reducing Electromagnetic Interference (EMI)

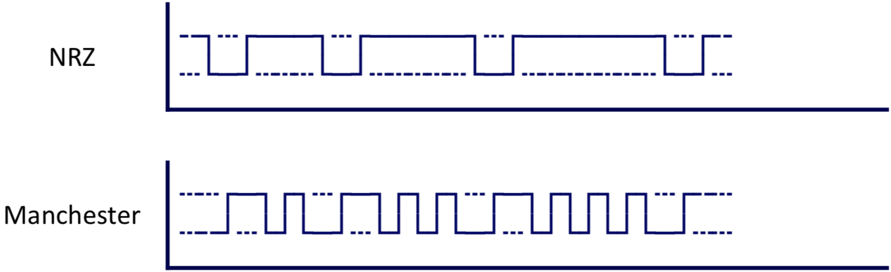
- Add shielding to wires
- Use twisted pair wiring
- Use coding with few rising/falling signal edges



Bit Coding: dealing with the clock

NRZ is less noisy

Manchester carries the clock with him on every single bit

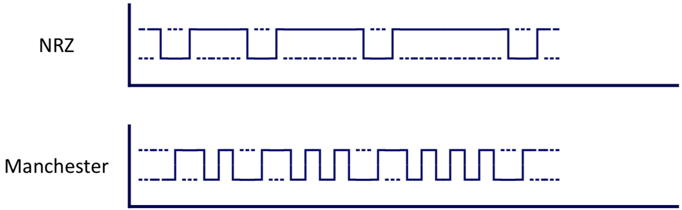


Clock drift problem!

Bit Coding: dealing with the clock

You know how a clock works, right?

- Clock drift is caused by natural variations of quartz (environment)
- Receiver must sample signal at right time instant
- Clock drift leads to de-synchronization (and bad interpretation of bit sequence -> errors)
- Bit timing has to be re-adjusted continually
- Commonly used: rising/falling signal edges



Bit Coding: NRZ Bit Stuffing

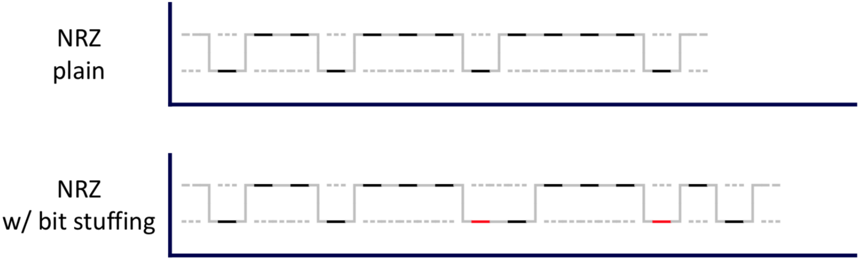
Problem

When using NRZ coding, sending many identical bits leaves no signal edges that could be used to compensate for clock drift

Solution

Insertion of extra bits after n consecutive identical bits

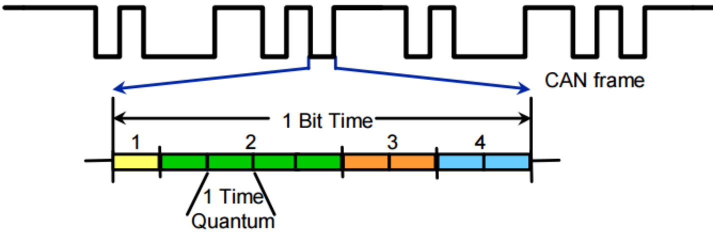
Example with stuffing width of 3 bits



Bit Coding: A Single Bit has a world inside him

A single bit is composed by:

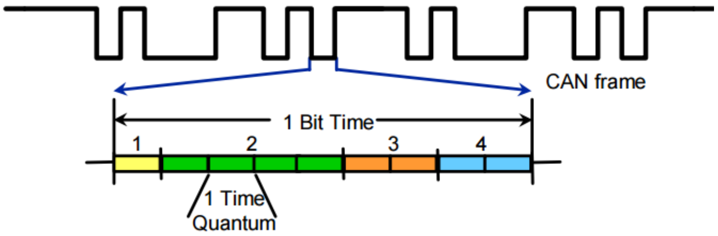
- 4 segments: each segment has a role (4 colors in the Fig)
- from 8 to 25 **Time Quanta** (TQ) (single slot in the Fig)
- TQs are generated by programmable divide of the Oscillator
- TQ is the smallest discrete timing resolution used by a CAN node



Bit Coding: A Single Bit has a world inside him

Some constraints:

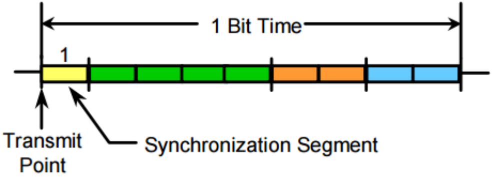
- Each segment is composed by an integer number of TQs
- Segments are non-overlapping
- TQs are generated by programmable divide of the Oscillator
- The bit time, and therefore the bit rate, is selected by programming the width of the Time Quantum and the number of Time Quanta in the various segments



Bit Coding: The First Segment

Synchronization Segment:

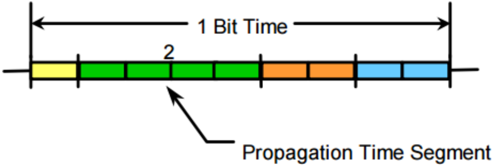
- The Synchronization Segment is used to synchronize the various bus nodes
- On transmission, at the start of this segment, the current bit level is output
- Bus State change (if any) expected to occur within this segment by the receivers
- The length of this segment is always 1 Time Quantum



Bit Coding: The Second Segment

Propagation Segment:

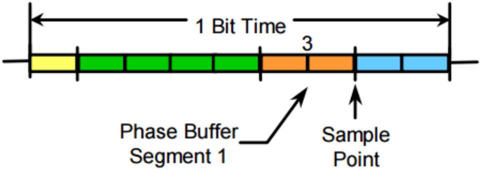
- Allows for signal propagation (across network and through nodes)
- Programmable length ([1-8] Time Quanta, 4 in the Fig)
- This is necessary to compensate for signal propagation delays on the bus line and through the electronic interface circuits of the bus nodes
- Do you think a long bus prefer 8 TQs or 1 TQ?



Bit Coding: The Third Segment

Buffer Segment One:

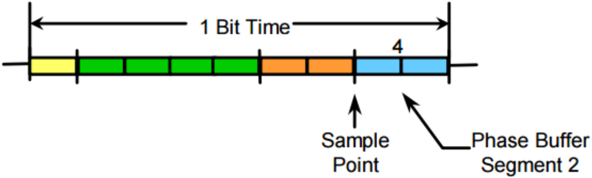
- This segment is where we "play the game"
- Allows for lengthening of bit during Re-synchronization
- Bus state is sampled at the end of this segment. The sample point is the point of time at which the bus level is read and interpreted as the value of the respective bit.
- Programmable length ([1-8] Time Quanta, 2 in the Fig)
- It is the only segment that can may be lengthened during resynchronization



Bit Coding: The Forth Segment

Buffer Segment Two:

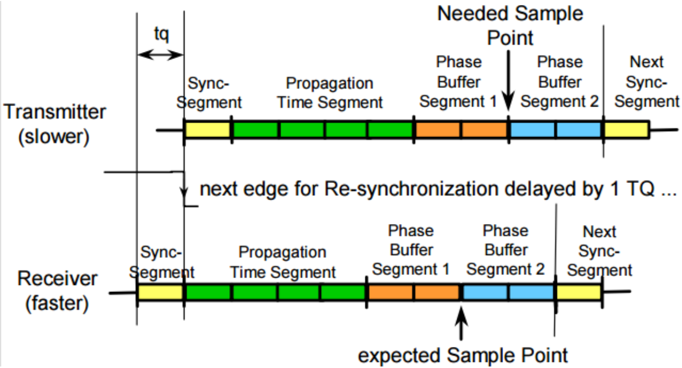
- This segment is where we "play the game"
- Allows for shortening of bit during Re-synchronization
- Programmable length ([1-8] Time Quanta, 2 in the Fig)
- The information processing time begins with the sample point and is reserved for calculation of the subsequent bit level. It is less than or equal to two Time Quanta long.
- The length of this segment must be at least as long as the information processing time



Bit Coding: Bit lengthening

The lengthening stage (the receiver is faster than the transmitter):

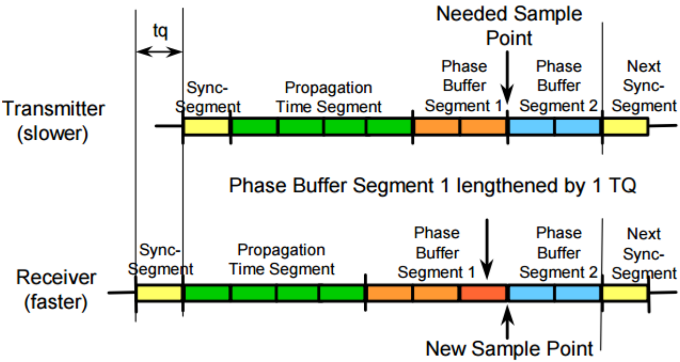
- Resynchronization: we need to compensate the oscillator tolerances within the different CAN nodes.
- In this example, the transmitter oscillator is slower than the receiver oscillator, the next falling edge used for resynchronization may be delayed. So Phase Buffer Segment One is lengthened



Bit Coding: Bit lengthening

The lengthening stage (the receiver is faster than the transmitter):

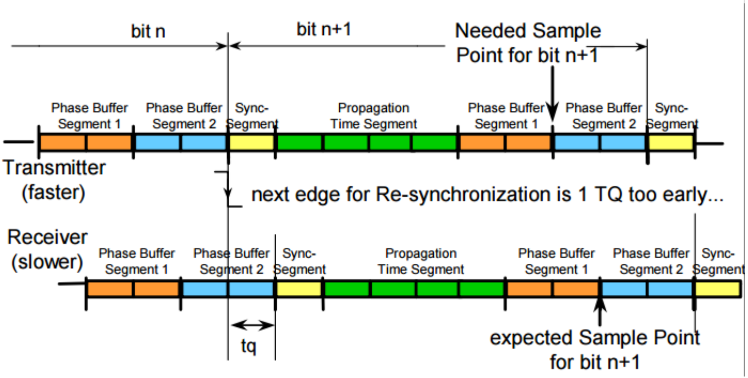
- In this example, the transmitter oscillator is slower than the receiver oscillator, the next falling edge used for resynchronization may be delayed. So Phase Buffer Segment One is lengthened...
- ...in order to adjust the sample point and the end of the bit time



Bit Coding: Bit shortening

The shortening stage (the receiver is slower than the transmitter):

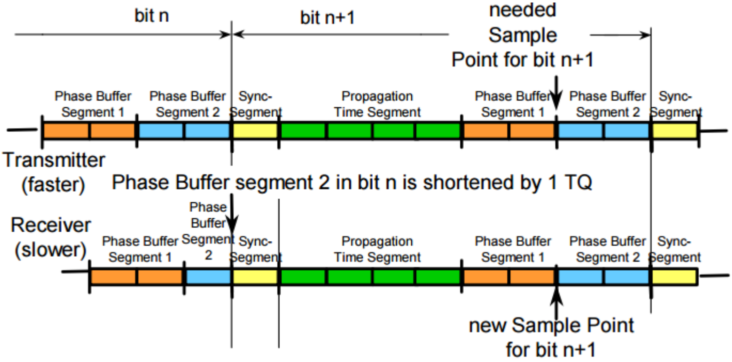
- In this second example, the transmitter oscillator is faster than the receiver oscillator, the next falling edge used for resynchronization may be too early. So Buffer Segment Two is shortened (during bit n)...



Bit Coding: Bit shortening

The shortening stage (the receiver is slower than the transmitter):

- In this second example, the transmitter oscillator is faster than the receiver oscillator, the next falling edge used for resynchronization may be too early. So Buffer Segment Two is shortened (during bit n)...
- ...in order to adjust the sample point for bit n+1 and the end of the bit time



Bit Coding: Jump Width

Jump Width

Amount by which bit length can be adjusted during Re-synch is defined as Synchronization Jump Width

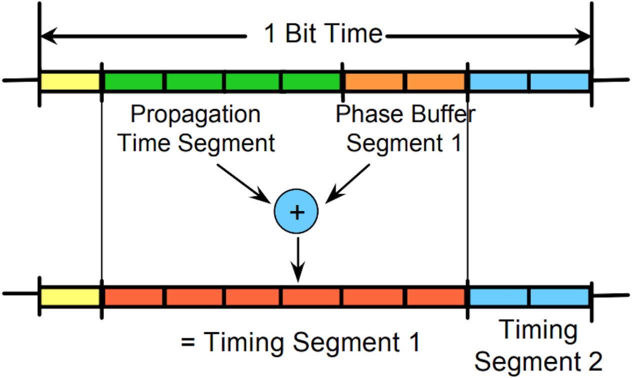
- N° of TQ by which Phase Buffer Segment One can be lengthened
- N° of TQ by which Phase Buffer Segment Two can be shortened

Programmability of Synchronization Jump Width is mandatory

- Minimum of 1 TQ, maximum of 4 TQ
- But it may not be longer than Phase Buffer Segment Two

Bit Coding: Bit Timing

For ease of programming many CAN Modules combine Propagation Time Segment and Phase Buffer Segment One (i.e. they only use 3 segments)

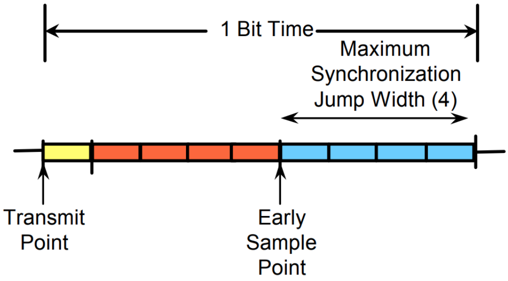


Phase Buffer Segment 2 is then known as Timing Segment 2

Bit Coding: Dynamic Sample Position (*early sampling*)

Why Program the Sample Position?

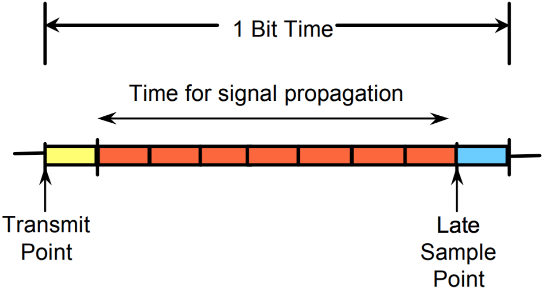
- Programming of the sample point allows "tuning" of the characteristics to suit the bus -> **flexibility**
- Early sampling -> more TQs in Buffer Segment 2 -> Jump Width of 4 TQs available
- Early sampling decreases the sensitivity to oscillator tolerances
- Lower cost oscillators (e.g. ceramic resonators) can be used



Bit Coding: Dynamic Sample Position (*late sampling*)

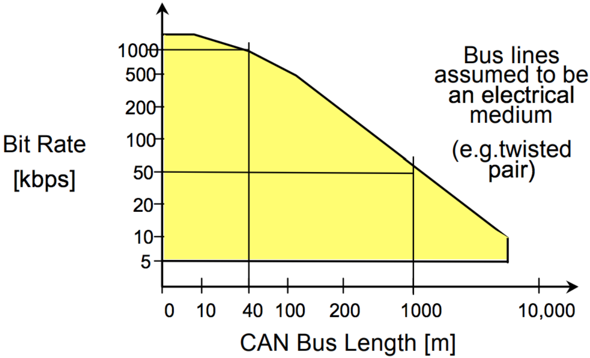
Why Program the Sample Position?

- Late sampling allows maximum signal propagation time -> **reachability**
- Maximum bus length and poor bus topologies can be handled
- More Time Quanta in the Propagation Time



Tradeoff between Bit Rate and Bus Length

Late sampling vs *early sampling* opens the tradeoff challenge



Up to 1 Mbps at 40m bus length (130 feet)