FlexRay

From Wikipedia, the free encyclopedia

FlexRay is an automotive network communications protocol developed by the **FlexRay Consortium** to govern on-board automotive computing. It is designed to be faster and more reliable than CAN and TTP, but it is also more expensive. The FlexRay consortium disbanded in 2009, but the FlexRay standard is now a set of ISO standards, ISO 17458-1 to 17458-5.[1]

FlexRay is a communication bus designed to ensure high data rates, fault tolerance, operating on a time cycle, split into static and dynamic segments for event-triggered and time-triggered communications.[2]

Features [edit]

FlexRay supports data rates up to 10 Mbit/s, explicitly supports both star and bus physical topologies, and can have two independent data channels for fault-tolerance (communication can continue with reduced bandwidth if one channel is inoperative). The bus operates on a time cycle, divided into two parts: the static segment and the dynamic segment. The static segment is preallocated into slices for individual communication types, providing stronger determinism than its predecessor CAN. The dynamic segment operates more like CAN, with nodes taking control of the bus as available, allowing event-triggered behavior.[3]

Consortium [edit]

The FlexRay Consortium was made up of the following core members:

- Freescale Semiconductor
- Bosch
- NXP Semiconductors
- BMW
- Volkswagen
- Daimler
- General Motors

There were also Premium Associate and Associate members of FlexRay consortium. By September 2009, there were 28 premium associate members and more than 60 associate members. At the end of 2009, the consortium disbanded.

Commercial deployment [edit]

The first series production vehicle with FlexRay was at the end of 2006 in the BMW X5 (E70),[4] enabling a new and fast adaptive damping system. Full use of FlexRay was introduced in 2008 in the new BMW 7 Series (F01).

Vehicles [edit]

- Audi A4 (B9) (2015–)[5]
- Audi A5 (F5) (2016–)[6]
- Audi A6 (C7) (2011-2018)[7]
- Audi A7
- Audi A8 (D4) (2010–2017)[8]
- Audi Q7 (2015-)
- Audi TT Mk3 (2014–)
- Audi R8 (2015–)
- Bentley Flying Spur (2013-2019)
- Bentley Mulsanne (2010–)[4]
- BMW X5 (E70) (2006–2013)[4]
- BMW X6 (E71) (2008–2014)[9]
- BMW 1 Series
- BMW 3 Series
- BMW 5 Series (2009–2017)[4]
- BMW 6 Series (2011–2018)[10]
- BMW 7 Series (2008–2015)[4]
- Lamborghini Huracán
- Mercedes-Benz S-Class (W222) (2013–2020)[11]
- Mercedes-Benz S-Class (C217) (2014–2020)[12]
- Mercedes-Benz E-Class (W213) (2016–2023)
- Mercedes-Benz C-Class (W205)
- Mercedes-Benz C-Class (W206) (2021–)
- Mercedes-Benz S-Class (W223) (2020–)
- Rolls-Royce Ghost (2009–)[4]
- Land Rover
- Volvo XC90 (2015–)[13]

Details [edit]

Clock [edit]

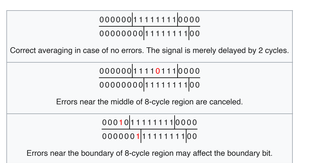
The FlexRay system consists of a bus and ECUs (Electronic control unit). Each ECU has an independent clock. The clock drift must be not more than 0.15% from the reference clock, so the difference between the slowest and the fastest clock in the system is no greater than 0.3%.

This means that, if ECU-s is a sender and ECU-r is a receiver, then for every 300 cycles of the sender there will be between 299 and 301 cycles of the receiver. The clocks are resynchronized frequently enough to assure that this causes no problems. The clock is sent in the static segment.[14]

Bits on the bus [edit]

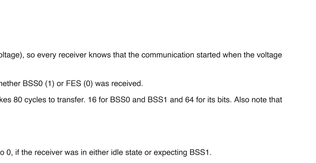
At each time, only one ECU writes to the bus. Each bit to be sent is held on the bus for 8 sample clock cycles. The receiver keeps a buffer of the last 5 samples, and uses the majority of the last 5 samples as the input signal.

Single-cycle transmission errors may affect results near the boundary of the bits, but will not affect cycles in the middle of the 8-cycle region.



Correct averaging in case of no errors. The signal is merely delayed by 2 cycles.

Errors near the middle of 8-cycle region are canceled.



Errors near the boundary of 8-cycle region may affect the boundary bit.

Sampled bits [edit]

The value of the bit is sampled in the middle of the 8-bit region. The errors are moved to the extreme cycles, and the clock is synchronized frequently enough for the drift to be small. (Drift is smaller than 1 cycle per 300 cycles, and during transmission the clock is synchronized more than once every 300 cycles).

Frame [edit]

All the communication is sent in the form of frames. The message consists of bytes $\{x_0, x_1, \ldots, x_{m-1}\}$, packed in the following way:

- Transmission Start Signal (TSS) – bit 0
- Frame Start Signal (FSS) – bit 1
- *m* times:
  - Byte Start Signal 0 (BSS0) – bit 1
  - Byte Start Signal 1 (BSS1) – bit 0
  - 0th bit of *i*-th byte
  - 1st bit of *i*-th byte
  - 2nd bit of *i*-th byte
  - ...
  - 7th bit of *i*-th byte
- Frame End Signal (FES) – bit 0
- Transmission End Signal (TES) – bit 1

If nothing is being communicated, the bus is held in state 1 (high voltage), so every receiver knows that the communication started when the voltage drops to 0.

The receiver knows when the message is complete by checking whether BSS0 (1) or FES (0) was received.

Note that 8-cycle per bit has nothing to do with bytes. Each byte takes 80 cycles to transfer. 16 for BSS0 and BSS1 and 64 for its bits. Also note that BSS0 has value 1, and BSS1 has value 0.

Clock synchronization [edit]

Clocks are resynchronized when the voted signal changes from 1 to 0, if the receiver was in either idle state or expecting BSS1.

As synchronization is done on the voted signal, small transmission errors during synchronization that affect the boundary bits may skew the synchronization no more than 1 cycle. As there are at most 88 cycles between synchronization (BSS1, 8 bits of the last byte, FES and TES - 11 bits of 8 cycles each), and the clock drift is no larger than 1 per 300 cycles, the drift may skew the clock no more than 1 cycle. Small transmission errors during the receiving may affect only the boundary bits. So in the worst case the two middle bits are correct, and thus the sampled value is correct.

Here's an example of a particularly bad case - error during synchronization, a lost cycle due to clock drift and error in transmission.

Errors that happened in the example:

- Because of a single-bit error during synchronization, the synchronization was delayed by 1 cycle
- Receiver clock was slower than sender clock, so receiver missed one cycle (marked X). This will not happen again before the next synchronization due to limits on maximum allowable clock drift.
- Because of a single-bit error during transmission, a bit was voted wrongly near the result.

Despite so many errors, the communication was received correctly.

The green cells are sampling points. All except the first are synchronized by the 1->0 edge in the transmission fragment shown.

Signal to be sent	1	0	1	0	1
Signal sent	11111111	00000000	11111111	00000000	11
On the bus	11111111	00000000	11111111	00000000	1011
Received	11111111	00000000	111111X1	00000000	1011
5-maj voted	111 1 1110	100 0 000	111 1 1100	000 0 1011	

Development tools [edit]

When developing and/or troubleshooting the FlexRay bus, examination of hardware signals can be very important. Logic analyzers and bus analyzers are tools which collect, analyze, decode, store signals so people can view the high-speed waveforms at their leisure.

The future of FlexRay [edit]

Ethernet may replace FlexRay for bandwidth intensive, non-safety critical applications.[15]

See also [edit]

- Byteflight
- Controller Area Network (CAN)
- Local Interconnect Network (LIN) - lower cost and lower bandwidth than CAN
- MOST Bus

References [edit]

1. ^ Lorenz, Steffen (2010). "The FlexRay Electrical Physical Layer Evolution" (PDF). *Automotive 2010*. Archived from the original (PDF) on 16 February 2015. Retrieved 16 February 2015.
2. ^ Vaz, R. M.; Hodel, K. N.; Santos, M. M. D.; Arruda, B. A.; Netto, M. L.; Justo, J. F. (2020). "An efficient formulation for optimization of FlexRay frame scheduling". *Vehic. Commun.* **24**: 100234. doi:10.1016/j.vehcom.2020.100234. S2CID 213291314.
3. ^ "How FlexRay Works". Freescale Semiconductor. Archived from the original on 23 February 2015. Retrieved 21 March 2014.
4. ^ a b c d e f Strobel, Otto (2013-02-28). *Communication in Transportation Systems*. IGI Global. p. 61. ISBN 9781466629776.
5. ^ "The driver assistance systems and integrated safety". *Audi MediaCenter*. Retrieved 2019-02-21.
6. ^ "Driver assistance systems". *Audi MediaCenter*. Retrieved 2019-02-21.
7. ^ Regler, Richard; Schlinkheider, Jörg; Maier, Markus; Prechler, Reinhard; Berger, Eduard; Pröll, Leo (2011). "Intelligent electrics / electronics architecture". *ATZextra Worldwide*. **15** (11): 246–251. doi:10.1365/s40111-010-0269-9. S2CID 107330814.
8. ^ "Audi Technology Portal - Networking". *Audi Technology Portal*. Retrieved 2019-02-21.
9. ^ "The BMW X6". *BMW Press Portal*. Retrieved 2019-03-08.
10. ^ "The new BMW 6 Series Convertible". *BMW Press Portal*. p. 32. Retrieved 2019-03-08.
11. ^ "2322446_83_Fahrwerk_S_Klasse_en.doc". *marsMediaSite* (in German). Retrieved 2019-03-08.
12. ^ "2480996_PI_Kurvenneigung_C217_ENG.docx". *marsMediaSite* (in German). Retrieved 2019-03-08.
13. ^ Fleiss, Michael; Müller, Thomas M.; Nilsson, Martin; Carlsson, Jonas (2016-03-01). "Fahrzeugintegration des Antriebsstrangs bei Volvo". *ATZ - Automobiltechnische Zeitschrift* (in German). **118** (3): 16–21. doi:10.1007/s35148-015-0202-7. ISSN 2192-8800. S2CID 183153508.
14. ^ "Introduction to FlexRay". *www.star-cooperation.com*. STAR ELECTRONICS. Archived from the original on 2016-12-20. Retrieved 2016-12-09.
15. ^ Hammerschmidt, Christoph (18 June 2010). "Beyond FlexRay: BMW airs Ethernet plans". *EE Times*. Retrieved 16 February 2015.

External links [edit]

- FlexRay Overview Technical Overview of FlexRay by National Instruments
- FlexRay Specification

v · t · e Technical and *de facto* standards for wired computer buses [show]
v · t · e Automation protocols [show]
Authority control: National Israel · United States
Category: Network protocols