

Related Protocols



This is the fifth in a series of articles by Jason Turner & Clinton Smith from

REDARC Technologies. They are writing about the latest in CAN and developments in this rapidly changing field. In the last article we discussed the higher level protocols that make up the CAN Application Layer. The CAN Application Layer determines what the messages actually are and what they mean. In this article we will discuss several Related Protocols as well as pondering what the future may hold for CAN.

ISO 11992

ISO11992 (often used by Trailer Electronic Braking Systems - TEBS) uses the CAN data link layer, however it differs from other CAN protocols as it uses an alternative physical layer. Most CAN applications use ISO11898 as the basis for their physical layer standard which is a 5V standard, however ISO11992 does not have a fixed voltage signal level, rather it defines low and high signals as being less than 1/3 and greater than 2/3 of a given reference voltage respectively. This is why there can be problems connecting a trailer with 12V

TEBS system to a 24V tractor, despite both using the same standard.

The ISO11992 Application Layer is based on J1939, with slight differences in the structure of the IDs and different message content.

Unfortunately the last commercially available ISO11992 transceivers were discontinued in November 2012, with no alternatives made available to date.

This means that development of any third-party ISO11992 based products is no longer possible.

SAE J1587/1708

J1708 describes a physical layer standard that is not CAN, but is electrically the same as RS-485 without terminating resistors.

J1587 is the Application Layer that is used on top of J1708. J1587 is often described as a predecessor to J1939, and is somewhat similar to J1939 in the way that messages are defined with almost all, if not all, MIDs (Message IDs) incorporated into J1939 with the MIDs matched to SPNs (Suspect Parameter Numbers).

In Figure 1 you can see an extract of the J1587 standard defining MIDs 51 and 52 alongside an extract of the J1939 standard defining SPNs 51 and 52. This comparison shows that these J1939 SPNs match the

J1587 MIDs, however, you may notice that J1939 uses metric measurements and different scaling.

LIN

The third related protocol we should briefly discuss is LIN (Local Interconnect Network). This is one protocol seen in heavy vehicles and it exists as a low-cost alternative to CAN. LIN is a single wire serial bus system (though it requires common ground), runs at a maximum of 20kbps, and operates with a master/slave configuration.

Being only a single wire, it can't tolerate as noisy environments as CAN, and running at such a lower baud rate means that it is not suitable for time-critical tasks.

However, as most of the protocol is defined in software, the LIN hardware is much cheaper than CAN, so it is commonly used for more simple, less important functions, such as electric mirrors, window lifts, wiper functions, seat warmers, etc.

For this reason, LIN is generally seen as a complement to CAN rather than a competitor. LIN works by the master sending out predefined headers on a set schedule, and when a slave node sees a header that it is supposed to respond to, it tacks its data onto the end of the header. All nodes observe the bus and record the data as necessary.

Figure 1 - J1587 vs J1939

SAE J1587 Revised JUL 1998

A.51 Throttle position - The position of the valve used to regulate the supply of a fluid - usually air or fuel/air mixture - to an engine. 0% represents no supply and 100% is full supply.

Parameter data length: 1 character
Data type: Unsigned short integer
Bit resolution: 0.4%
Maximum range: 0.0 to 102.0%
Transmission update period: 0.2 s
Message priority: 3
Format: PIO Data
51 a
a- Throttle position

A.52 Engine Intercooler temperature - The temperature of liquid found in the engine intercooler, located after the turbocharger.

Parameter data length: 1 character
Data type: Unsigned short integer
Bit resolution: 1.0° F
Maximum range: 0.0 to 255.0° F
Transmission update period: 1.0 s
Message priority: 4
Format: PID Data
52 a
a- Engine intercooler temperature

SAE J1939 Revised JAN 2008

SPN 51

Engine throttle position - The position of the valve used to regulate the supply of a fluid - usually air or fuel/air mixture - to an engine. 0% represents no supply and 100% is full supply.

Data length: 1 byte
Resolution: 0.4 %/bit, 0 offset
Data range: 0 to 100 %
Operational range: Same as data range
Type: Measured
Supporting information: PGN 65266

SPN 52

Engine Intercooler temperature - Temperature of liquid found in the intercooler located after the turbocharger.

Data Length: 1 byte
Resolution: 1° C/bit, -40° C offset
Data Range: -40 to 210° C
Operational Range: Same as data range
Type: Measured
Supporting information: PGN 65262

CAN - what does the future hold?

The evolution of vehicle technology has meant that CAN in its current form is becoming less suitable for modern tasks. Daimler/Mercedes Benz have been using CAN in their cars since the 1991 S-Class was released, and in trucks since the 1992 SK heavy class truck was released. Since then communications requirements have increased dramatically and CAN alone hasn't been able to provide the required speed and bandwidth needed for modern vehicles. This has meant that other technologies have had to be employed alongside CAN, with the current generation of vehicles implementing both LIN and FlexRay as well. These protocols all communicate through a central gateway that is fast reaching its maximum throughput ability.

In future, Daimler/Mercedes predict replacing the central gateway with an automotive Ethernet backbone and utilising not only CAN, LIN, FlexRay and Ethernet, but also a new protocol called CAN with Flexible Data-rate' (CAN-FD). CAN-FD is the latest extension to the CAN protocol, where the bit rate is increased during the data phase of the CAN message and also allows up to 64 bytes to be transmitted instead of the traditional 8 bytes. The increase in bandwidth and payload means that CAN-FD should be suitable for a further 1-2 vehicle generations.

CAN-FD

'CAN with Flexible Data Rate' is the most significant change to the CAN standard since the extended frame format was introduced in 1995. As just mentioned, at its heart CAN-FD is simply CAN with a greater data payload and faster bit rate, but (I) how is this achieved whilst retaining the robustness CAN is famous for? And (II) is it backwards-compatible with traditional CAN?

Before we answer those questions, perhaps the first question that should be asked is why change the CAN protocol at all? The answer to this question was touched upon briefly when discussing Daimler/Mercedes' need for more speed and bandwidth. Three decades ago when Bosch started development of the CAN protocol, eight bytes of data transmitting at one megabit per second (Mbps) was ample for almost all needs in a vehicle, but since then needs have changed. Car entertainment systems have been introduced, many devices that were previously fully mechanically controlled have now become electronically

controlled, and even electronic devices that were previously operating in isolation have now been integrated into the main vehicle network. With so many more devices communicating to one another and bigger data payloads needing to be transmitted, traditional CAN is neither fast enough nor has a high enough throughput to suit most modern vehicle requirements.

Bosch recognised this fact, and began work on changing the CAN protocol to be able to handle higher payloads and transmit at greater speeds without losing all the benefits CAN has over competing protocols. This meant that this new breed of CAN still had to be cheap, very robust, scalable, and familiar to current CAN users and implementers. After successful internal trials, Bosch presented its new CAN-FD protocol at the 13th International CAN Conference in 2012. Before long they had won the support of almost all major automotive manufacturers, as well as silicon manufacturers; two things that would be critical for the successful adoption of CAN-FD. As a result, all of the major silicon manufacturers have begun development of CAN-FD capable CAN controllers, which are predicted to eventually supersede traditional CAN controllers in the same way that CAN 2.0b (Extended CAN) controllers superseded CAN 2.0a (Standard CAN) controllers when it was introduced in 1995. Unfortunately though, at the time of writing, there is a distinct lack of low-end microcontrollers with CAN-FD capable controllers currently in development. Silicon manufacturers appear to be focussing on high-end microcontrollers for the roll out of CAN-FD and whilst large devices such as ECUs will use high-end micros, vehicles contain many small sensors and devices that need to communicate on the same bus and must use low-end micros due to size and cost restraints. It is thought that more

low-end CAN-FD capable micros will begin to appear once the first CAN-FD micros go into full production.

Since CAN-FD controllers are expected to replace traditional CAN controllers, this brings us back to the earlier question of whether or not CAN-FD is backwards compatible with traditional CAN. As with most things of this nature, the answer to this is "sort of". CAN-FD has been designed to accommodate traditional CAN, but certain data link layer differences mean it is simply not possible to transmit CAN-FD messages on a CAN bus with any devices that are not "CAN-FD aware". This being said, all CAN-FD enabled devices are capable of operating in traditional CAN mode, meaning that a CAN-FD device can be connected to a traditional CAN bus, but it will be restricted to transmitting only traditional CAN messages. However, once all devices on a CAN bus have been updated with CAN-FD compatible controllers, then they can simply enable CAN-FD mode and utilise the benefits of this new type of CAN.

The other question we asked earlier was how does CAN-FD achieve higher bit rates and greater data throughput without compromising CAN's impressive list of advantages over other protocols? The name of this new development presents the answer - "flexible data rate".

What this means is that unlike traditional CAN that has one fixed data rate, CAN-FD can actually have two, one for the arbitration phase of the message and a different one for the data phase (see Figure 2). For CAN to achieve non-destructive bitwise arbitration, one of the main reasons for its robustness, all devices on the bus must be kept in sync. This is the main contributor to the limits on bus length and bit rates, as bits must be able to travel the full length of the bus in a fraction of a bit time to stay in sync,

Figure 2 - CAN-FD Arbitration Phase vs Data Phase

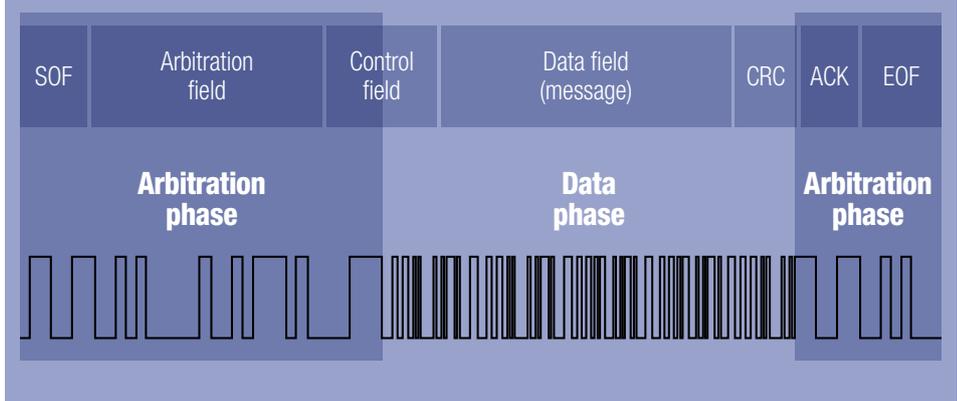


Figure 3 - CAN-FD payload lookup table

DLC	Number of bytes
0-8	0-8
9	12
10	16
11	20
12	24
13	32
14	48
15	64

so longer busses mean further to travel and faster bit times mean less time to make the journey. But CAN devices only need to be in sync during the arbitration phase, as bitwise arbitration does not occur during the transmission of the data payload itself.

CAN-FD utilises this fact to increase the overall bit rate by keeping the bit rate the same as traditional CAN during the

Figure 5 - CRC Polynomials

CRC	Polynomial
CRC-15	$(x^{15}+x^{14}+x^{10}+x^8+x^7+x^4+x^3+1)$
CRC-17	$(x^{17}+x^{16}+x^{14}+x^{13}+x^{11}+x^6+x^4+x^3+x^1+1)$
CRC-21	$(x^{21}+x^{20}+x^{13}+x^{11}+x^7+x^4+x^3+1)$

For this to occur, the data phase bit rate must be at least eight times the arbitration phase bit rate.

As discussed in article #3, the data length code (DLC) field of a CAN message is four bits long which allows a range of 0 to 15 unique values. With traditional CAN, any DLC greater than or equal to eight is simply considered to be eight; with CAN-FD, the values 9 to 15 are used to represent 12 to 64 byte payloads as described by a lookup table (Figure 3).

Of course to accommodate these new features there had to be some changes to the data link layer (Figure 4); the traditional RTR bit was reappropriated as a CAN-FD

The last significant change to the CAN frame format is the length of the CRC polynomial.

Due to the longer data fields possible with CAN-FD, new CRC polynomials had to be used to ensure that the Hamming Distance was maintained at six; or, in other words, the CRC is guaranteed to detect up to five bit flipped bits in the message. This results in 15-bit CRCs for traditional CAN frames, 17-bit CRCs for CAN-FD messages with payloads up to sixteen bytes, and 21-bit CRCs for CAN-FD messages with payloads greater than sixteen bytes (Figure 5). Further to this though, there is also a change to how bit-stuffing is handled in the CRC field.

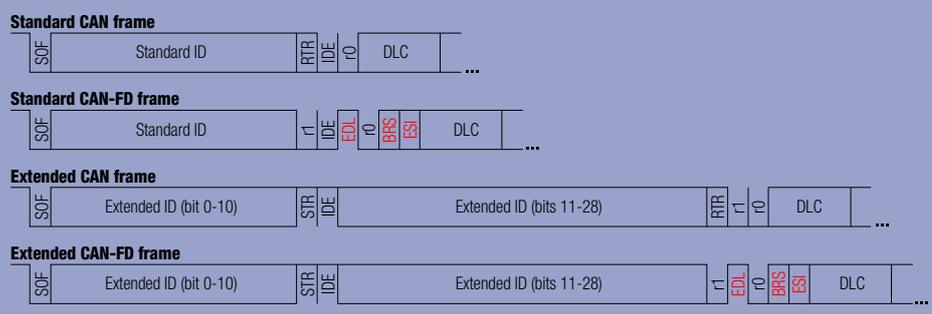
Previously, bit-stuffing was handled the same as it was elsewhere, with a bit toggle immediately succeeding five bits of the same polarity; this means that the actual number of bits transmitted/received could vary depending on how many sequences of successive bits with the same polarity occurred.

Also, since the stuff bits weren't part of the CRC calculation, two consecutive bit errors at the edge of a bit stuffing sequence could potentially go undetected. To prevent this, CAN-FD has introduced fixed bit-stuffing into the CRC sequence, where a stuff bit is inserted after every four sequence bits regardless of whether the preceding bits were all of the same polarity or not.

In article #6 we will continue to explore CAN FD and its applications. We will also consider applications in special vehicles, and introduce the reader to AUTOSAR.

For more information on CANbus protocols or other CANbus matters contact Jason Turner at Redarc Technologies on 08 8322 4848, visit redarc.com.au or send an email to power@redarc.com.au

Figure 4 - CAN-FD Frame



arbitration phase and then ramping up the bit rate significantly during the data phase.

After the CRC is sent, the bit rate drops back to the original rate and all devices resynchronise before the acknowledge bit is sent by the receiving devices.

Increasing the net bit rate allows each message to be sent faster, but it is only half of the solution. The second main feature of CAN-FD is that the maximum data payload per message is increased from eight bytes to a more reasonable sixty-four bytes.

Furthermore, because the data phase is the part of the message that does not require synchronisation and can therefore utilise the higher bit rate, it is possible that a much larger CAN-FD message can actually be sent in less time than a smaller traditional CAN message.

indicator bit called EDL (Extended Data Length) and shifted right of the r1 bit, this means that remote frames are no longer possible in CAN-FD.

Since very few CAN HLPs use remote frames, and remote frames can still be sent using a traditional CAN message if required, it was simply not necessary to be included with CAN-FD.

Another change is the addition of the BRS (Baud Rate Switch) bit inserted to the right of the r0 bit, and used to indicate if the faster bit rate is to be used for the data phase or not.

Directly following the BRS bit is another new bit called the ESI (Error State Indicator) flag. This flag is used to show if a transmitting node is in the error active state or the error passive state.