



# CAN, or Controller Area Networks, the basics

This is the first in a series of articles by Jason Turner & Clinton Smith from Redarc. Our readers will find their information very informative. We have asked them to bring us up to speed covering the latest in CAN and developments in this rapidly changing field.



Jason Turner & Clinton Smith are both degree qualified Electronics Design

Engineers from REDARC Technologies Pty Ltd that design, develop and manufacture smart electronic products for the Automotive Industry. Both Jason & Clinton in their work at REDARC have developed significant experience in Control Area Network (CAN) training and consulting, module design, development and manufacturing.

In November 2013 the 14th International CAN Conference (ICC) was held in Paris, France. REDARC has attended every ICC since 2005 and Clinton and Jason were the engineers representing Redarc at the 14th ICC. The conference covers many CAN-based topics with presenters and attendees representing companies from all aspects of CAN development, from chip-makers to system designers, automotive OEMs, tool designers, and of course end-of-line product designers.

Let's first explain the basics about CAN, or Controller Area Networks.

In our experience many people are still unaware of CAN and the impact it is having on the industry. To explain what CAN is, it is best to pose a question. Have you heard of 'the internet'? Yes we hear you say. The internet is what can be considered a WAN, or Wide Area Network, as it is a giant network of devices all connected together over a wide-area, hence the name Wide Area Network.

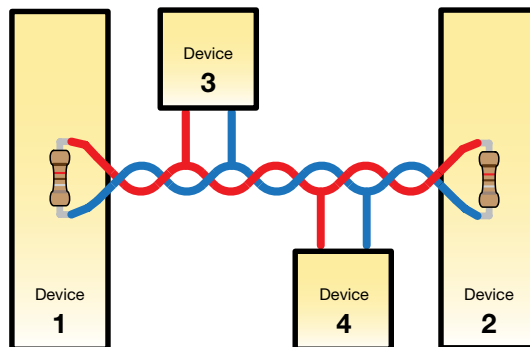
CAN can be thought of in a similar way. Put simply it is a smaller network, made up of devices that are primarily controllers for controlling other things, thus the name Controller Area Network. In the case of vehicles, these controllers may be things like an Engine ECU (Electronic Control Unit, that is the brains of the engine), or perhaps the transmission ECU, lighting control unit, ABS controller, or even the dashboard controller.

But CAN isn't just found in vehicles, it is commonly found in everything from washing machines to Olympic rings. Believe it or not, your home is full of CAN busses, and if your car was manufactured after 2005 it's practically guaranteed to have at least one CAN bus.

So what makes CAN special? Why is it worth writing about? Well, there are a number of reasons:

- It is low cost, needing only two wires with a couple of resistors on the ends.
- It is robust and can handle harsh conditions that other networks can't – under the bonnet of a vehicle for example.
- It is fast (compared to other serial protocols) with rates up to one megabit per second.
- It simplifies installation and therefore reduces the amount of wiring needed, saving time, copper, fuel and hence money.
- The hardware is standardised, giving it a high level of connectivity, meaning that certain CAN products can be purchased off-the-shelf and simply plugged in.

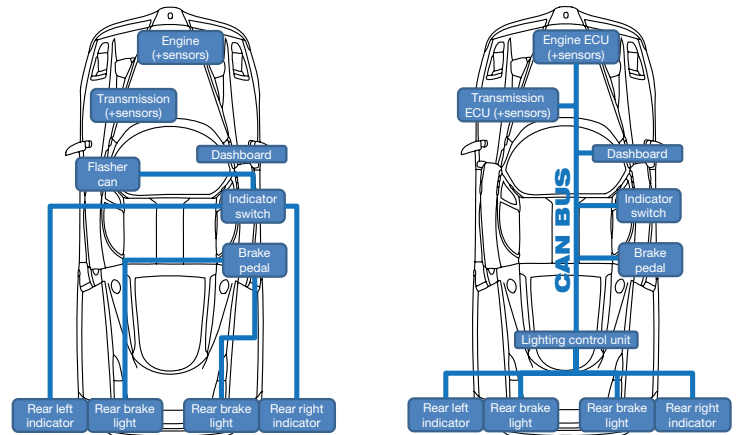
A CAN bus in simple terms is just a twisted pair of wires with a resistor at either end. If that's all it was, then it wouldn't be much use, so obviously it's the way these wires are used that makes CAN special. Whilst the physical bus is very simple, the protocol transmitted on the bus is not, in fact it's very clever. Since this is an introduction to CAN we won't bore you with too much technicality and move onto what this cleverness actually achieves.



Pictured above is a CAN bus. There are two devices (device 1 & 2) on it from the

beginning however further devices can be added simply by connecting them in parallel (device 3 & 4). If one device sends a message, all devices on the bus will see that message and they can then choose to listen to it or ignore it. Essentially, devices decide which messages they'll listen to and which ones they'll filter out because they don't care about them. In this way – if we return to our internet comparison – it's a bit like Twitter. Someone sends out a tweet and the whole world can see it, yet only those 'following' that tweet will bother reading it, while everyone else will ignore it. This is different to conventional point-to-point systems where messages or signals were sent directly to only one recipient. So how does this help? Well let's take a look at an example of traditional wiring in a vehicle.

If we take a look at how a vehicle is conventionally wired (pictured below left), we can see that every switch requires a direct connection to every output it uses; that every sensor needs a direct connection to every device that needs its information (such as the dashboard). Wiring in this way can be messy, uses a lot of copper, and can lead to bulky and complex wiring harnesses.



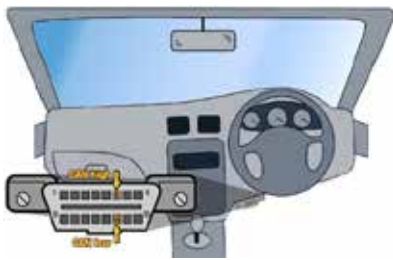
Conversely, if we look at the wiring of a vehicle that utilises a CAN bus (pictured above right), we can see that each device only requires a single connection to the bus as all information is shared over the network. For example, when an indicator switch is turned on, the lighting control unit sees the change in state and turns the indicator light on. At the same time, the dashboard sees the change and displays the indicator lamp to the driver. Same goes for the transmission, when a gear is changed, both the dashboard and the engine see this change and respond accordingly.

The added benefit is that if multiple devices need the same information (e.g. engine temperature) each device does not



require a separate sensor; only one sensor is required for all devices on the bus. This means costs can be cut by reducing the number of sensors required, the amount of copper required, and the labour involved in wiring the system; without any negative impacts on functionality. Furthermore, the complexity of the wiring harness is reduced so isolating cabling issues can become much easier.

Practically speaking there are other advantages as well. For example, since all the information from all the different sensors are now available on the one bus, debug tools can access all this information at once to try and diagnose problems. You will no doubt be aware of the On Board Diagnostics connector (or OBD-II) that has been mandatory in all cars sold in the US and Europe for well over a decade; what you might not know is that there are multiple protocols available that are allowed to be used on that connector, but only one is mandatory for all of these cars – that one is CAN.



The amount of information that is available on a vehicle's CAN bus can be quite extensive and can include parameters such as: engine speed, fuel pressure, throttle position, coolant temperature, barometric pressure, oil temperature, fuel economy, actual engine torque, engine run time, turbocharger RPM, injection timing, vehicle speed, and many many others.

Having access to all this information is invaluable for diagnostics, but can also be extremely useful for other applications as well. To highlight this we will now discuss a few practical examples of how this information is useful.

The first example is a product that we designed and manufactured for one of the major truck OEMs who had some prototype trucks that they were building. These trucks have an additive called AdBlue that is required to be added to the fuel to ensure the trucks meet emission standards; however, since these were prototype trucks the dashboards had not yet been updated to include a gauge to show the AdBlue tank level. Since this information was available on the CAN bus, we were able to build a device that would read the information on the bus, display a gauge for the AdBlue tank, sound an alarm when the level was low, and display other



requested information such as current gear and selected gear.

The second example relates to another major truck OEM that was installing a brand new 18-speed autoshift transmission into one of their trucks that had previously only ever had had a manual transmission. At first the transmission did nothing as it needed more information about the engine to work, so we were able to get it functioning correctly by sending the information it needed over the CAN bus. Of course that was just the first step, while the transmission was now working there was quite a bit of fine tuning required to make these gear shifts faster and smoother than an experienced human driver and for this we then needed to return to the CAN bus. An example that required fine tuning was when a fully laden truck attempted to drive up a very steep incline; the problem was that the transmission needed to shift down gears as the vehicle slowed, but when it disengaged the driveline to do the change, the driveshaft slowed down faster than the engine so it could never make the shift until it gave up and dropped back into first or second gear. Essentially what this meant was that if the hill was too steep the truck would almost stop halfway up and then basically have to do a hill start from there. We were able to overcome this problem by monitoring parameters on the CAN bus such as whether or not an up shift of down shift was taking place, when the driveline was engaged or disengaged, the engine speed and the shaft speed; and from all those parameters we could determine when we needed to rapidly slow the engine to make a fast shift. When the conditions were right, we'd send messages to the engine brake and/or the exhaust brake to slow down the engine to the correct amount, and the result was a very fast and smooth shift with the vehicle barely dropping speed at all. We were also able to do the reverse by firing an output to the countershaft brake relay when upshifting and going downhill, again resulting in super smooth shifts.

These are just two examples of how the information on a CAN bus can be used, but the possibilities are almost endless. Hopefully this has demonstrated how useful the CAN bus can be. As we've discussed, it's like the internet for vehicles;

except for one important caveat. The CAN bus specification only defines the physical layer and what we call the data-link layer. That is, it defines how data is communicated over a CAN bus, but it does not define what the data actually is. This is an important point to make, because this one thing leads to a lot of confusion for some people; people often think all CAN busses are created equal, but they are not. Because the hardware layer and the data-link layer are standardised in the CAN specification, it means we can take a CAN diagnostic tool and plug it into any CAN bus and extract data from it – but what it does not mean is that we can understand the data!

Think of it in terms of language: what the CAN standard defines is like the letters of the alphabet and how to pronounce them. Since we all speak English, we can pick up an English book and understand what the words and phrases that are constructed with these letters mean. However, if we pick up a French book that has words and phrases constructed of exactly the same letters (and we don't speak French) then we would not be able to understand what it is saying without a French dictionary to interpret it. The same goes for CAN. We can extract the data from any CAN bus, but we can only interpret the data if we have the correct 'dictionary'. When talking about CAN, these different 'languages' are known as 'Higher Level Protocols' and a number of them are open standards, so we can simply purchase the standard (or 'dictionary') and go ahead and interpret the data. For example, trucks all use an open standard called SAE J1939. This standard is documented in detail so there is very little problem with reading and interpreting the different CAN busses on almost any truck. Unfortunately it is not as easy with passenger vehicles as manufacturers define their own proprietary higher level CAN protocols.

This means that for most passenger vehicles you will have little chance to interpret the data on the vehicles. That being the case there is one thing that means at least partial interpretation of these CAN busses is possible. As stated above, all US and European sold cars made after 2008 must have OBD-II and it must have CAN. This CAN bus uses a standardised higher level protocol, and therefore the majority of the information on that bus can be read and interpreted.

In our next article we will delve into the detail regarding CAN for the experienced technician. Subjects covered will include physical and data-link layers, bit stuffing, arbitration, CAN data frame and CAN error states to name a few.

[www.redarc.com.au](http://www.redarc.com.au)  
or [power@redarc.com.au](mailto:power@redarc.com.au)