mantracourt
Advanced Intelligent Instrumentation

TECHNICAL BLOG
INTRODUCTION TO DIGITAL DEVICES PART 2

OCTOBER 2016

DR. CLIVE VALLANCE, SENIOR ELECTRONICS ENGINEER
Mantracourt Electronics

Mantracourt Electronics Ltd
The Drive, Farringdon, Exeter, EX5 2JB, UK
W: mantracout.com

Contact our Technical Sales Team
T: +44 (0)1395 232020
E: sales@mantracourt.com

Mantracourt Electronics Ltd
The Drive, Farringdon, Exeter, EX5 2JB, UK
W: mantracout.com

Contact our Technical Sales Team
T: +44 (0)1395 232020
E: sales@mantracourt.com

INTRODUCTION

The previous instalment focussed on the physical communication interface between digital devices and a receiving station. This time I would like to step up a layer and look at the information that is sent along the physical wires. This is often described as the communication protocol and is essentially the 'language' that is used to communicate.

I would like to maintain our analogy with the mobile phone interface a little longer. Recall last time we had two phones that were able to communicate and send audio information. This is akin to the physical wires in a digital conditioner. This audio connection does not care about whether the two individuals at each end of the phone understand each other, only that the connection is made. So, this extends a little further if we introduce a language that is being used to communicate. As long as the language is the same, the two parties are able to communicate and understand one another. This is exactly the same in the digital conditioner scenario.

Of course in our discussion we have not spoken about how people negotiate the use of the physical connection. If you have ever made a conference call you will understand how the complexity of getting onto that communication channel increases with the number of participants. If two people try and speak at the same time there is a collision, the information is lost or garbled and we instinctively cease communication and retry at a later point. In extreme circumstances someone may be allocated to control the access of the rest of the group to the communication channel which removes the chance of a collision. Alternatively, there may be one person requesting information from a number of participants in turn. All of these possibilities of negotiating access to a channel exist with a digital communication scenario.

In digital systems there is often one master who controls the communications between entities. This relationship is often referred to as master slave and is itself a mechanism of stopping collisions in the data. These systems often rely on the fact that there is only one master. Another arbitration scheme used in CAN relies on the priority level of a message. This is a little more involved but to be as simple as possible any message on a CAN BUS has a priority level. When a message is sent the message with the highest priority is allowed to transmit data by winning the arbitration phase. This is non-destructive and in a well designed system eliminates collisions. The CAN arbitration method does mean that a master is not required to control the access to the BUS. There are other access schemes which involve allocated time slots and tokens but I won't dwell on those at this time.

At Mantracourt we maintain communication languages for different purposes and applications. Some are quick and others are easy to understand without a translating tool. Some are industry standardised and others are proprietary. In general we provide 5 main protocols.

Mantracourt Electronics Ltd
The Drive, Farringdon, Exeter, EX5 2JB, UK
W: mantracout.com

Contact our Technical Sales Team
T: +44 (0)1395 232020
E: sales@mantracourt.com

MODBUS RTU

ModBUS is a standardised protocol originally designed for communication between programmable logic controllers. It is versatile and robust and commonly used for industrial communication systems. It is simple to implement, openly published and royalty free. ModBUS RTU is probably the most commonly implemented variant of ModBUS and is used for serial communications. ModBUS TCP is now gaining momentum and is used across Ethernet networks although the real time performance is slightly more limited. ModBUS in general is quite efficient in its use of the communications channel as it is a binary protocol. Unfortunately it is not the easiest to read without a software tool to translate it. The protocol itself operates on a request response basis, so it is an example of our master slave implementation. We at Mantracourt offer digital devices that use ModBUS RTU on 232 or 485 serial interfaces.

I would also like to bring your attention to something we often come across in Modbus tools. Although it is standardised, the offset applied when parameter locations are read is a little ambiguous. I have come across numerous incarnations of PC and logic controller interfaces and they do have a tendency to vary the offset applied to a parameter. This is often the result of confusion between number starting from zero or starting from one. To be honest the pros and cons of zero and one addressing is a rant that I think I should keep for another day.

In conclusion, when the offsets are resolved the data across these networks is delivered reliably, robustly and efficiently. It is a protocol that I like to use. I think the greatest difficulty comes with readability which is often overcome with these translating tools.

MANTRA-ASCII

This proprietary protocol is designed for simplicity of integration and debugging. The data can be connected through to a PC and displayed without any special software. A simple terminal program allows the user to evaluate the operation quickly and easily. This does come at the cost of overhead. The transmission of ASCII characters is not as efficient as the binary based protocols and as such it is limiting in terms of throughput. It is able to operate in two modes. Primarily the interface is request response, but there is also a streaming mode that exists. This then reduces the overhead in communication to increase the data rate but it does limit the operation to only a single sensor on the physical channel. As such it is not used often and has its limitations. Mantra-ASCII is available for use on 232 or 485 serial interfaces.

In conclusion the Mantra-ASCII communication is easy to debug and evaluate. As long as the data rate is not particularly

Mantracourt Electronics Ltd
The Drive, Farringdon, Exeter, EX5 2JB, UK
W: mantracout.com

Contact our Technical Sales Team
T: +44 (0)1395 232020
E: sales@mantracourt.com

high (circa 120 sample per second) it is a very simple and reliable protocol.

MANTRA-BUS

MantraBUS is one of our proprietary binary protocols designed to be efficient and flexible. It has much less overhead than the ASCII protocol and as such uses the physical layer bandwidth more efficiently. This leads to greater throughput on busy BUS interfaces. Of course as a manufacturer it is very useful because we have control over the interface and optimise it for our applications. This also means that it is much easier for us to support.

Unfortunately in most cases proprietary protocols do not play well with others (i.e. all the devices on the bus should communicate using the same language). As such system integrators have a choice. In the event that they are designing the entire system using the BUS then a proprietary protocol is fine and the protocol designer is able to offer lots of support. The flip side of this is that when the BUS is shared a standardised protocol is required and all the devices must use that same protocol.

CANOPEN

CANopen is a data format that may be used on a CAN BUS. Often people are unsure of the difference between the CAN physical and protocol layers as the lines are a little blurred. CAN open is maintained by the CiA (CAN in Automation) and is one of a number of higher layer CAN protocols. It essentially adds network management controls on top of the CAN network. It is gaining wider usage within industry but does tend to be restricted to certain industry applications at the moment.

Our greatest struggle with CAN protocols is that each of the hardware vendors put together their own proprietary interface for a PC. As such configuration software, including our own, is often limited to specific hardware interfaces. There is another separation between CAN and the other BUS technologies that ought to be raised here. CAN is slightly separated from our analogy of phones and languages. We discussed briefly what the complexities are of gaining access to the BUS and how CAN systems achieve this. The fact that you have a CAN device will mean that it will cohabit on a CAN BUS and not cause any damage to the communications. However this does not mean that out of the box the system nodes will understand one another or know what they need to communicate and to whom. Effectively you will need to tell each node who it reports to and how that is done. This is where all the knowledge and expertise of the integrator is required as the efficiency of the entire system depends upon this configuration stage.

MANTRA-CAN

Mantra-CAN is our CAN based proprietary protocol designed to be flexible for end users. The key benefit to this is that it can be

Mantracourt Electronics Ltd
The Drive, Farringdon, Exeter, EX5 2JB, UK
W: mantracout.com

Contact our Technical Sales Team
T: +44 (0)1395 232020
E: sales@mantracourt.com

configured to deliver data onto CAN networks in a format chosen by the user. This is also a limitation as it can never be fully compliant with other protocols but is great at delivering data onto proprietary networks. Essentially it is a protocol designed to play nicely with others without really interacting fully. One thing to note is that the configuration of the device must take place before connecting to the BUS.

We have customers who swear by it and others who are nervous about designing a system that uses it. In general this all comes down to understanding what this kind of protocol can and cannot do. If you understand what it is designed to do, its great.

In summary, there are binary, ASCII and CAN protocols available. Each application may require one or other of these protocols. The first thing to understand is that if speed is important to you, a binary protocol is probably the way forward and if you just want to plug in and use a terminal interface then it is probably easier to use the ASCII type interface. I personally believe that the protocol is chosen for each scenario but I understand that many vendors like to choose one and specialise in it.

There are two aspects I think you should take from this if nothing else. When using digital conditioners, before you get to the software you are using, ensure the physical wires and the protocol are the same between all of the devices you are connecting together. This is where 80% of the problems arise in digital implementations.

Next time I would like to move higher up through the layers again and look at the user interface software. This can be completely customised, use our interface drivers or even our free software. It all depends upon the complexity of the application but I will try and take you through a few options.

For more detailed advice and support please contact us at mantracourt.com

This article was written by Dr. Clive Vallance, Senior Electronics Engineer, Mantracourt Electronics Ltd