TECHNOTES - Technology Note

August 2008

PROFINET Naming and TCP/IP addressing

If we compare an office Ethernet environment with what is common on the factory floor there are huge differences. Not only on the physical side, but also how the addressing and the traffic work. In this paper we would like to give some inside information how device addressing is organized in PROFINET.

When you come in the office in the morning and you boot up your PC during your first cup of tea or coffee, one of the things it does is getting itself an IP-address from the networks DHCP-server. You get this IP-address, which looks like 192.168.250.10, for as long as you are connected. Tomorrow it could be different.
That you get a different address each time is not a problem as you initiate network traffic to a known source, like the fileserver or the web.

But in a machine you must be sure that the device you communicate with is the correct one. Not that the same command opens a valve on one day, and starts a motor the next day. And then there is the issue of device replacement. If a device breaks down it must be easy to replace and reconfiguration must be limited to a minimum. It must be possible for our friend Heinz, the plant's maintenance guy who only knows screwdrivers, to replace a device at three o'clock at night.

A little sidetrack to how PROFINET communication functions.
In all industrial communication protocols there are two types of communication:
1. Configuration and diagnostics.
2. I/O data exchange.

PROFINET uses normal TCP/IP communication for the configuration and diagnostics tasks. Therefore each device needs to have a TCP/IP address.
The I/O data exchange does not use TCP/IP communication. It is very low level Ethernet communication. Whole TCP/IP is bypassed and PROFINET is working directly on the Ethernet controller. The reason to do this is that the PROFINET-IO frame has less overhead in the protocol handling and becomes smaller. This increases the throughput because less computing power is needed.

So we want to make a device unique on the network and do TCP/IP communication with it. Normal industrial practice is to use a couple of rotary switches on the device to set the address. And a rotary switch is something Heinz understands although he needs a very small screwdriver. But in PROFINET the number of devices is unlimited. Then you'd need an unlimited number of rotary switches.

Therefore PROFINET implemented the Name mechanism. To make a device unique it is given a name, a string that is loaded in the device. A PC-software tool is normally used, incorporated in the Network configuration tools.
It sends out a query message on the network to which all PROFINET devices react. In the tool a list appears of all devices discovered with their MAC-ID, their default name and a default IP-address. The MAC-ID is the low-level Ethernet-address of a device.
Probably there are several devices, e.g. SmartSlice I/O stations, which have the same default name out-of-the-box.

Now to find which device is where, there is the option "blink". You select one of the devices found and activate the "blink" function. Then the device will start to blink its LEDs. Now it is just a matter of sending out Heinz to find the blinking device. He will confirm a blinking device and its position. With the position is probably a name or code associated and that is what you program in the device. You have then uniquely named the device.

The names of the devices you now remember and you enter them in the configuration of the PROFINET IO-controller. Plus an IP-address that is needed to do configuration or any other TCP/IP communication (FTP, WEB, etc.).

Now the PROFINET IO-controller knows the names of the devices it needs to communicate with. When a PROFINET IO-controller starts, it asks on the network: „Hello, is there someone with this name“. The device with this name replies and the PROFINET IO-controller then issues the device its IP-address. Now the device is ready to receive its parameters and configuration, then the IO-control cycle starts.

Then the shit doesn't hit the fan but the IO-device. It is three o'clock at night. Heinz runs to his warehouse with spare equipment and gets out a replacement device. Remember there are no rotary switches to turn but the device needs to get the proper name. It is stored in the old device. But how does Heinz get the name in the new device? Vendors implemented different ways. But it all comes down to storing the name in something that can be taken out of the old one and put in the new one. There are devices with SD-cards or smaller memory cards. But remember it is Heinz who has to handle these memory cards. What if it drops on the floor? And the floor is three stories down?
Other possibility is USB-sticks but everybody can use them. Like SD-cards. And where are they after 7 years of operation?

Our opinion is that the name storage should be part of the functioning device and be recognizable as such. For our SmartSlice system the name storage function is put in a so called memory end-plate, called GRT1-END-M. When you first assign the name to a SmartSlice station, the name is stored in the bus coupler, and automatically copied into the memory end-plate

Heinz replaces the faulty buscoupler with a spare one. The buscoupler will notice there is a GRT1-END-M with a stored name and will start using that name. The PROFINET IO-controller will find the buscoupler with the correct name and starts communication with it. Problem solved. Machine is running again and Heinz can return to his cup of tea and Sudoku.

SmartSlice's automatic name backup and restore make device replacement quite easy in PROFINET.

René Heijma
Product Engineer Industrial Communication
Omron Europe