CAN OVER ETHERNET GATEWAYS - A CONVENIENT AND FLEXIBLE SOLUTION TO ACCESS LOW LEVEL CONTROL DEVICES

G.Thomas, D.Davids, CERN, Geneva, Switzerland
O.Holme, ETH Zurich, Switzerland

Abstract

CAN bus is a recommended fieldbus at CERN. It is widely used in the control systems of the experiments to control and monitor large amounts of equipment (IO devices, front-end electronics, power supplies). CAN nodes are distributed over buses that are interfaced to the computers via PCI or USB CAN interfaces. These interfaces limit the possible evolution of the Detector Control Systems (DCS). For instance, PCI cards are not compatible with all computer hardware and new requirements for virtualization and redundancy require dynamic reallocation of CAN bus interfaces to different computers. Additionally, these interfaces cannot be installed at a different location than the front-end computers. Ethernet based CAN interfaces resolve these issues, providing network access to the field buses. The Ethernet-CAN gateways from Analytica (GmbH) were evaluated to determine if they meet the hardware and software specifications of CERN. This paper presents the evaluation methodology and results as well as highlighting the benefits of using such gateways in experiment production environments. Preliminary experience with the Analytica interfaces in the DCS of the CMS experiment is presented.

INTRODUCTION

The control systems of CERN require the monitoring and control of a wide range of front-end equipment, much of which is installed in technically challenging locations with exposure to ionising radiation, strong magnetic fields and significant sources of electrical noise. A range of fieldbuses, including CAN bus [1], have been evaluated and selected by CERN to enable robust communication with the front-end hardware, leading to their widespread adoption in the LHC accelerator and experiments. As the readout computers and corresponding bus interfaces must be located outside of the hostile environment, the field buses are often more than 100 metres in length.

In the case of CAN bus, the communication on the bus is commonly driven by an OPC Data Access (OPC DA) server that runs on a computer and exposes the monitored hardware process variables and commands via the standard OPC interface protocol. The OPC server enables any OPC compliant client to connect and access the front-end data, independently of the details of the low level protocol used on the CAN bus for each type of hardware. A standard CERN solution for building the supervisory control system layer is to use the commercial SIMATIC WinCC Open Architecture (WinCC OA) [2] control system toolkit which natively supports OPC communication.

CURRENT STATUS AND LIMITATIONS

At the time of the construction of the LHC accelerator and experiment control systems, the connection between the control system computers and CAN buses was typically provided by internal PCI or PCIe cards or standalone USB connected interfaces. These CAN readout systems worked reliably throughout the first running period of LHC.

Changes in the computers running the control system software have been a significant motivating factor to re-evaluate the choice of CAN interface. For instance, the use of blade servers offers opportunities to minimize the rack space used by control system computers. The most compact blade servers do not provide the possibility to add PCIe expansion cards and therefore the PCIe-CAN interfaces are no longer a viable option in this context.

Due to the ubiquity of USB interfaces on modern computer hardware, the USB-CAN interfaces continue to provide a method of connecting CAN buses to any server. However, the limitation of maximum USB cabling length means that the computer and USB-CAN interface are typically installed either in the same rack or closely neighbouring racks. So with both PCIe and USB interfaces, the server and CAN bus interface must be co-located requiring that the control software servers are installed in or near to the racks where the CAN buses currently terminate. This imposes strong limitations on the possible re-organisation of rack hardware. With the current USB or PCIe interfaces, any significant change of server location would involve the costly installation of new CAN cabling.

Furthermore, there is increased interest in the use of redundancy at the control server level and the use of server virtualisation. Using the redundancy mechanism offered by WinCC OA, each control application runs on two computers. For robustness against power failures and network issues, it is beneficial to locate the two servers in different physical locations. As both computers need to access the front-end hardware, each server must be able to connect to the CAN bus interface, which makes it impossible to rely on the existing PCIe CAN or USB CAN devices used previously. Likewise, in the case of server virtualisation, the physical location of the server that runs the control software is not necessarily known in advance so it is essential to have a CAN interface which can be contacted from any location, independently of the distance between the server and interface hardware.

Offering network access to the CAN interface by Ethernet delivers a solution to these issues. This concept was demonstrated at CERN in the TOTEM and CMS experiments by coupling existing USB-CAN interfaces to Ethernet connected USB hubs. Using this approach the

ISBN 978-3-95450-148-9

CMS DCS was able to relocate and reduce the server rack space and to implement control server redundancy, without needing to change the existing CAN cabling. From operational experience it became apparent that, while providing the required functionality, the readout chain with both the Ethernet-USB and USB-CAN conversions was complex and difficult to debug in case of problems. For this reason, CERN initiated an evaluation to find a direct Ethernet-CAN interface that would meet the existing and future requirements of the organisation.

The evaluation identified the Anagate Ethernet-CAN Gateway family of devices from Analytica GmbH [3] as a candidate for a standard CERN Ethernet-CAN interface. These devices are based on Linux running on ARM CPUs and are currently available in two generations of hardware, the first being based on a CAN controller chip from Microchip and the second using an FPGA to drive the CAN buses. The first generation hardware is available in models with 1, 2 and 4 CAN ports, packaged in stand-alone boxes. Additionally there is a 1U rack mountable unit which can contain up to three 4-port devices, giving a maximum of 12 CAN ports. The second generation hardware is DIN rail mountable and is offered in 2, 4 and 8 CAN port variants.

INTERFACE EVALUATION METHODS

It is essential to have stable CAN interface hardware in any production environment because any failure will prevent communication between the front-end hardware and the control system software. This can generate downtime of the experiments and creates frustration amongst the support teams. To minimise the issues and purchase the best hardware that would meet the end user's requirements, the following methods were used to evaluate the Anagate Ethernet-CAN gateways.

Defining CAN Interface Test Specifications

The first priority was to define a test specification list describing the objectives, procedures and pass/fail criteria to check that the Ethernet-CAN interface met the user requirements. The tests were grouped into 4 categories; namely Functionality, Usability, Performance and Stress. Only Performance and Stress test categories are described here in detail. The primary goal of this exercise was to define a unit test for each category that specified its objective, a testing procedure and the pass condition. This test case catalogue could then be used to evaluate the hardware and to report the test results.

- Performance Testing Objectives

One of the first questions when evaluating a new hardware is inevitably going to be: “How does it perform?” Alternative questions might be: What maximum throughput can it sustain? Can it handle high load? How fast does it respond? There are many different ways to ask the question but all of them refer to the performance of a product. In order to be able to answer these questions and to determine how effectively it can perform under our environmental conditions and constraints, a unit test specification was defined per performance test criteria. These unit tests included: throughput, endurance and latency. Throughput unit tests were devised to measure the maximum number of CAN frames per second the interface could handle per CAN bus before CAN frames started to be discarded by the interface. Tests were performed for each possible baud rate. Endurance unit tests involved running the interface with a significant load over a long duration to ensure that the throughput and response time did not degrade with long-term sustained load. Latency unit tests were designed to characterise the delay associated with sending CAN data through the interface hardware.

- Stress (resilience) Testing Objectives

It is also important to know how well the hardware behaves and recovers beyond its normal operation conditions. As the CAN interface is a key element in the readout chain, it is essential that it can recover quickly and automatically from failure conditions. It should not require a physical human intervention to act on the interface, since the hardware is distributed over distances of several kilometres and located in areas that are not always fully accessible during the operation of the experiment.

Stress testing involved putting the CAN hardware under exaggerated levels of stress to evaluate its stability in the non-ideal conditions encountered in the production environment. Three types of major failures were identified in order to assess the robustness of the interface and check how well it was able to recover.

Power failure unit tests involved powering off and on the interface via hardware and or software reboot during periods of active CAN communication and then analysing how the interface recovered from this state.

LAN failures tests consisted of making a physical disconnection of the network equipment (i.e. Ethernet cabling, Network switch, etc…) for varying durations during operation and analysing the recovery mechanism.

CAN network failure tests aimed at acting on the CAN bus to simulate errors such as putting a CAN port into BUS OFF and verifying that the interface could reset the CAN port without any negative impact on the system.

Writing of Software Tools

As described earlier, the control and monitoring data of the devices on the CAN buses is handled by WinCC OA, which communicates via a native OPC client to a vendor specific OPC server. This OPC server then communicates over CAN with the physical device concerned. The OPC servers were originally written for a CERN defined set of CAN interfaces, mostly from Kvaser AB [4], making use of the vendor specific hardware APIs.

To be able to use and evaluate the Anagate Ethernet-CAN gateways within the CERN control context, it was necessary to write a wrapper DLL that would map between the original vendor (Vendor 1) API used in the OPC server and the new hardware API from Analytica (Vendor 2). This is illustrated in Fig. 1. From the OPC server point of view, the wrapper DLL behaves in exactly the same way as the

Vendor 1 API, avoiding the need to rewrite parts of the OPC server when changing CAN interface hardware.

The wrapper DLL was implemented in C as a re-entrant multi-threaded library.

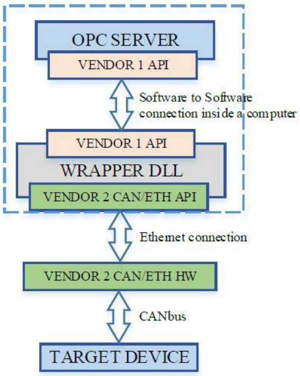


Figure 1: OPC-wrapper-CAN API software layout.

As the Vendor 1 API accesses the hardware via the CAN-port number and has no notion of how to access the Anagate hardware over the Ethernet network, a static configuration file is needed to associate hardware CAN-port numbers with the Anagate IP-addresses and IP-port numbers. To overcome the three types of major failures described in the stress testing section, a complex recovery mechanism was implemented in the wrapper DLL such that those problems become transparent for the OPC servers.

To facilitate the unit testing, a wide-ranging application for interactive testing was written in C. The test application was developed using the Kvaser API and can therefore use the wrapper DLLs to adapt to other CAN interfaces. The test tool can be called from batch scripts, enabling the automation of the required performance and stress tests. Additionally a set of PERL scripts were implemented to assist with the analysis of data generated during the tests.

The test application allows the execution of individual commands or scenarios of consecutive commands. For each CAN port, it is possible to instantiate independent sending and receiving threads. The traffic pattern of the sending thread can be controlled via three parameters: the number of CAN frames per burst, the time between bursts and the duration of the test in terms of the total number of bursts as shown in Fig. 2.

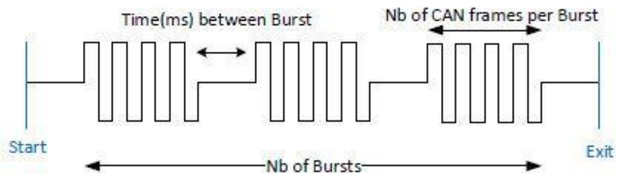


Figure 2: CAN frames transmit pattern.

An additional feature of the test application is that it can uniquely label CAN frames and record the delay between sending and receiving a given frame. By sending data on one CAN port and receiving it on a second port, it is possible to measure the latency of the round trip to the CAN bus and back, passing twice through the software and hardware interface layers. The latency statistics are written to file in the form of a frequency distribution of the delays.

TESTING AND RESULTS

The testing of the Anagate Ethernet-CAN gateway lasted for several months, with the aim of evaluating a maximum number of software and hardware aspects. Firstly, this involved checking the quality of the interface in terms of hardware parts (i.e. quality of CAN connectors, interface housing, etc…) and software elements (i.e. ease of installation, API readability, quality of documentation, etc…). Secondly, the unit tests described in the test specification document were carried out. Thirdly, the interfaces were installed in an experiment production environment to evaluate the stability and scalability over a period of several weeks with hundreds of CAN nodes distributed over several CAN buses. Throughout the testing period, the quality and responsiveness of the support from the vendor were evaluated.

Hardware and Software Product Inspection

After purchasing, the Anagate gateways were installed in the lab and the following criteria were evaluated:

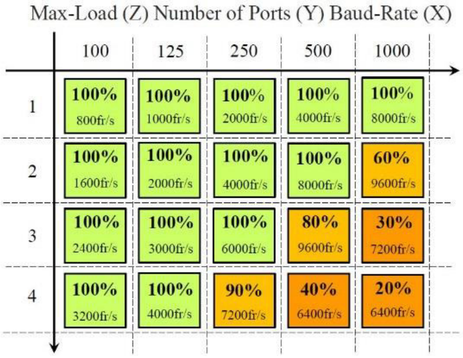
- The quality of the housing and the integrated connectors (CAN ports, internet and powering) indicated that the hardware was robust.
- The ease and rapid installation and configuration of the interface demonstrated a documentation and software of good quality
- Testing of the native API methods could be rapidly evaluated with little programming effort.

Performance Measurements

Knowing the maximum throughput the interface could handle in receiving and transmitting CAN frames (before CAN frames were discarded) was the first performance criteria to be measured.

To achieve this, we used two Windows 64-bit computers and 3 Anagate CAN Quattro gateways mounted in a 1U case. We installed the hardware in a rack and connected two of the four CAN ports of the first Quattro gateway (which was the device under test) to each of the other two Quattro gateways. For each CAN baud rate (10kb/sec up to 1Mbit/sec) both receive and transmit CAN traffic was generated and passed through the device under test in order to determine the maximum CAN load before frames started to be discarded. The results are shown in Fig. 3.

The findings correspond to tests with a duration of five minutes. The maximum load percentages are the thresholds at which no frames are discarded. When those thresholds are exceeded, frames may be discarded by the Anagate hardware. As can be seen, the maximum load percentage diminishes for high CAN baud rates and when all CAN buses are used. These limitations correspond to the maximum throughput of the Anagate gateway that is consistent with data provided by Analytica.



Max-Load (Z) Number of Ports (Y) Baud-Rate (X)

	100	125	250	500	1000
1	**100%** 800fr/s	**100%** 1000fr/s	**100%** 2000fr/s	**100%** 4000fr/s	**100%** 8000fr/s
2	**100%** 1600fr/s	**100%** 2000fr/s	**100%** 4000fr/s	**100%** 8000fr/s	**60%** 9600fr/s
3	**100%** 2400fr/s	**100%** 3000fr/s	**100%** 6000fr/s	**80%** 9600fr/s	**30%** 7200fr/s
4	**100%** 3200fr/s	**100%** 4000fr/s	**90%** 7200fr/s	**40%** 6400fr/s	**20%** 6400fr/s

Figure 3: Throughput measurement results.

Latency Measurements

Several tests under different network topologies (with routers and switches) were performed to quantify the latency introduced by the Anagate hardware. The latency measured includes the time it takes to send and read back a CAN frame across the various layers (software, hardware and LAN).

The results in Figure 4 show the distribution of latency introduced by the Anagate hardware in comparison to a USB-CAN interface, when operating with low load. The average latency of the Anagate interface is slightly higher, but more significantly, the latency distribution is wider, with a long tail including longer delays.

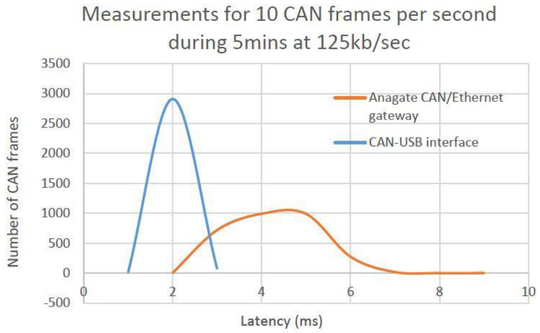


Figure 4: Latency measurements results.

Stress Testing Results

The Anagate interfaces were evaluated against several failure types that are encountered in a production environment. With recovery features included in the wrapper DLL, the Anagate interfaces automatically recovered from different type of failures such as power cut, network disconnection and CAN bus errors.

CMS Production Environment Tests

To test the Anagate hardware in a production environment, the Analytica interfaces were installed in CMS and integrated into the DCS in place of the previous USB-CAN interfaces. Tests covered the main types of CAN-based hardware to be controlled in CMS, namely power supplies and VME crates from WIENER Plein & Baus GmbH [5] and Embedded Local Monitor Boards (ELMBs) [6] designed by a collaboration between CERN, NIKHEF and PNPI. Large CMS CAN installations were selected to test the Analytica interfaces in the most demanding applications. Each test in Table 1 was executed for one week to evaluate performance and stability.

Table 1: CMS CAN Bus Test Applications

Hardware type	**Devices**	**Buses**	**Baud rate**
Wiener VME Crate	88	8	500 kb/s
Wiener Power Supply	136	10	100 kb/s
ELMB	104	8	125 kb/s

Initially, the Wiener OPC server encountered problems due to the additional latency of the Ethernet-CAN interface. The extra latency caused delays in the hardware response to OPC server requests, exceeding the expected round trip time and causing OPC item data to be marked as invalid. After discussion with Wiener, the problem was solved by modifying the OPC server to tolerate a user configurable additional latency in the readout chain.

The week long tests ran smoothly, indicating that the Anagate interface can be successfully integrated as a reliable component in the CMS DCS. An additional long term test was performed where the Wiener VME crate system ran for four months without any negative impact on the control system, in which time, more than 54 billion CAN frames were processed by the interface.

CONCLUSION

The Anagate Ethernet-CAN gateways from Analytica passed the unit tests including those which evaluated the functional, performance and robustness criteria. Furthermore, the tests in CMS demonstrated that the CAN interface could operate robustly in a production environment and that it provided the required quality of service for the experiment control system.

Throughout the testing period most of the issues reported to Analytica were analysed and fixed within a period of few weeks which demonstrated a very satisfactory level of collaboration and support.

The Anagate gateways offer a viable alternative to PCI and USB interface types, which enables the evolution of the CERN control systems towards virtualization and redundancy. There are ongoing studies for the integration of this interface into future OPC Unified Architecture based CERN control systems.

REFERENCES

[1] Q. King et al, “Fieldbuses for Control Systems at CERN”, June 2013, https://edms.cern.ch/document/1262875
[2] WinCC OA; http://www.etm.at/index_e.asp
[3] Analytica; http://www.anagate.com/index.html
[4] Kvaser AB; http://www.kvaser.com
[5] Wiener Electronics for Research & Industry; http://www.wiener-d.com
[6] Open Hardware Repository: CERN ELMB; http://www.ohwr.org/projects/cern-elmb/wiki

ISBN 978-3-95450-148-9