BE. services

REAL-TIME Test Framework for Industrial Control Systems.

Test the real-time performance of your Industrial Control Systems during the different phases of your development.

Real-time Test Framework

- Assists in the Industrial Control System (ICS) development process by measuring real-time performance at all steps.
- Ideal for ICS with the CODESYS V3 Runtime System using Real-time Ethernet communication for fieldbus like EtherCAT, PROFINET, POWERLINK and for Redundancy Systems
- Perfect match for controllers based on Linux

Real-time systems

Real-time systems can be found from very simple micro-controllers in embedded systems to highly sophisticated, complex and distributed systems. Real-time and embedded systems are broadly used in modern industrial processes and those rely on their good functioning. The areas of use are very wide, ranging from industrial automation control to equipment used in the health, automotive, airplane and aerospace industries.

While most of the Control Systems in the field run with a cycle time of several milliseconds, more demanding applications require not only a quicker cycle time (typically in the 1 ms range) but, and more importantly, a deterministic behavior. Most of the real-time systems perform safety or critical processes which may lead to fatality if not performed properly and in time.

This requires intensive testing in order to assure the proper execution of an application and testing should accompany the development process – at all stages, continuing in the early life period of the product.

The main challenges for the tests are

- Performing an accurate testing of the real-time performance of the industrial control systems before their release
- Detection of rare, sporadic, unexpected real-time performance issues
- Reducing the amount of manual work

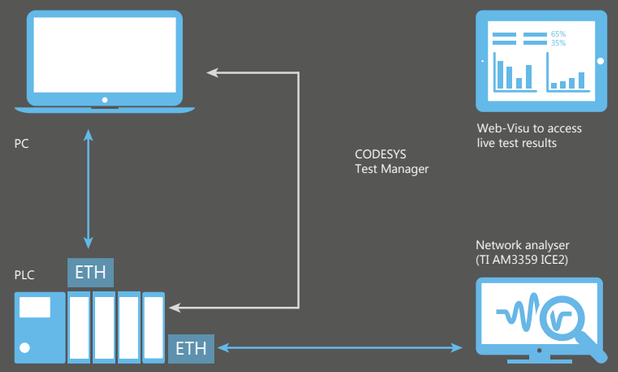
Our RT Test Framework helps you in your development process since

- It covers several typical and complex conditions for Industrial Control Systems. These can even be extended to cover customer specific ones.
- It is built in the well-known CODESYS user interface, making it easy to operate.
- It is compatible with any CPU / OS platform as it is run in the CODESYS RTS allowing the tests to be performed on almost any hardware platform.
- It is fully automated which makes it possible to run at each development milestone, allowing to locate problems easily.

Benefits

- Reduce costs thanks to a fully automated execution of the test cases. Once setup is complete, you simply need to launch the test and you get both live results in our attractive environment and a final complete and extensive test report.
- Gain operation time using the commonly used CODESYS Development System as User Interface. You do not need to get familiar with a new user interface since the test project comes as a CODESYS application containing visualization screens.
- Gain development time with the progressive testing at each phase of the development allowing you to compare real-time performance between implementation of code. You can easily narrow down the cause of the performance changes which simplifies debugging and optimizes maintenance time.

Architecture of the test system



Test system architecture

The tests are executed as a CODESYS project on a test PC.
The controller already contains a CODESYS Runtime System which supports SysEthernet and SysTimeGetUs.
The network analyzer (Texas Instruments AM3359ICE2) is used as I/O module.
For a more convenient and complete solution, we recommend installing CODESYS TestManager from 3S-Smart Software Solutions.

Testing scope

Using a hard real-time operating system does not guarantee the required performance. This can be caused by a bug in the firmware, complex conditions, task priority inversion, Ethernet driver not optimized, etc...

Test mechanism:

In order to control the real-time capability of the Industrial Control System, the following needs to be tested and calculated:

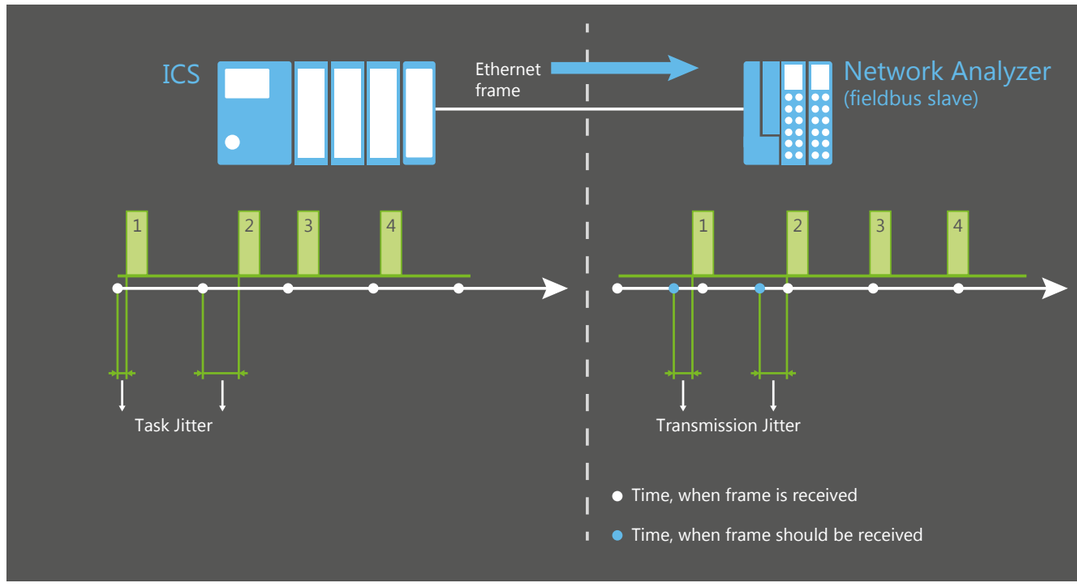
1. Task jitter
2. Ethernet frame transmission jitter
3. Ethernet frame roundtrip time

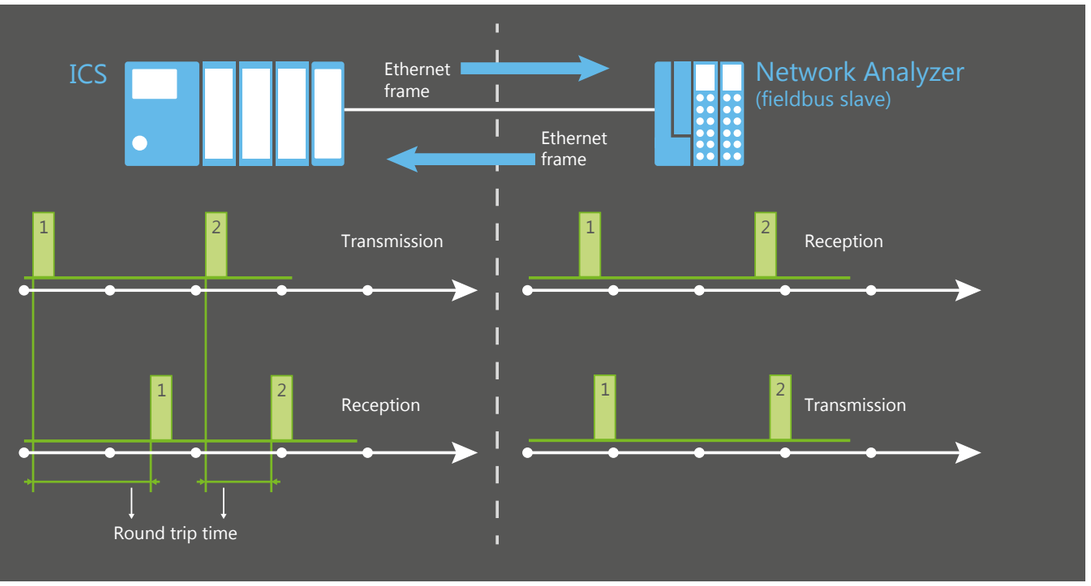
Test application

- The test application is executed in the ICS and sends Ethernet frames via the selected Ethernet interface cyclically. Each frame carries a timestamp (in microseconds) containing the exact time when the frame is sent.
- A Network Analyzer receives these frames, adds a receive timestamp and, as soon as the defined amount of frames is received, sends them back to the ICS. The frames are processed with a hard real-time Network Analyzer resulting in an extremely small jitter at reception.
- The ICS collects the send/receive frame results and calculates the maximum task and slave receive jitter. Slave receive jitter (corresponding to the ICS transmission jitter) is a significant value for the real-time characteristics of the system and shows how stable the ICS transmits the frames.

- In order to estimate the stability and performance of the ICS Ethernet frames reception, round trip test is executed.
- Frames are sent cyclically and each frame carries a timestamp (in microseconds), containing the time, when the frame is sent.
- A network analyzer receives the frame and sends it back immediately.
- The tested ICS receives the frame and calculates the time between the sending and reception of the frame – called roundtrip time (Troundrip).

- Physically, such a frame trip only takes a short time (about 100 us), but due to delay in the ICS communication stack, the reception of the frame may be postponed and this delay can be unstable.
- In real-time systems, the roundtrip time must be stable.
- A short roundtrip time means high performance of Ethernet communication – i. e. fast fieldbus I/O update.
- Roundtrip test can be activated optionally.





Test cases

- TEST_Normal. Real-time test is executed without any special test conditions
- TEST_NVL. During the test, network variables are transmitted in both directions
- TEST_TCPClient. The tested ICS communicates with a TCP Client
- TEST_DataServer. The DataServer communicates with the tested ICS via UDP
- TEST_FileAccess. During the test, file operations are executed
- TEST_OPC. During the test, plenty of tags are read from the tested ICS using OPC Client
- Additional test cases can be added to the test framework

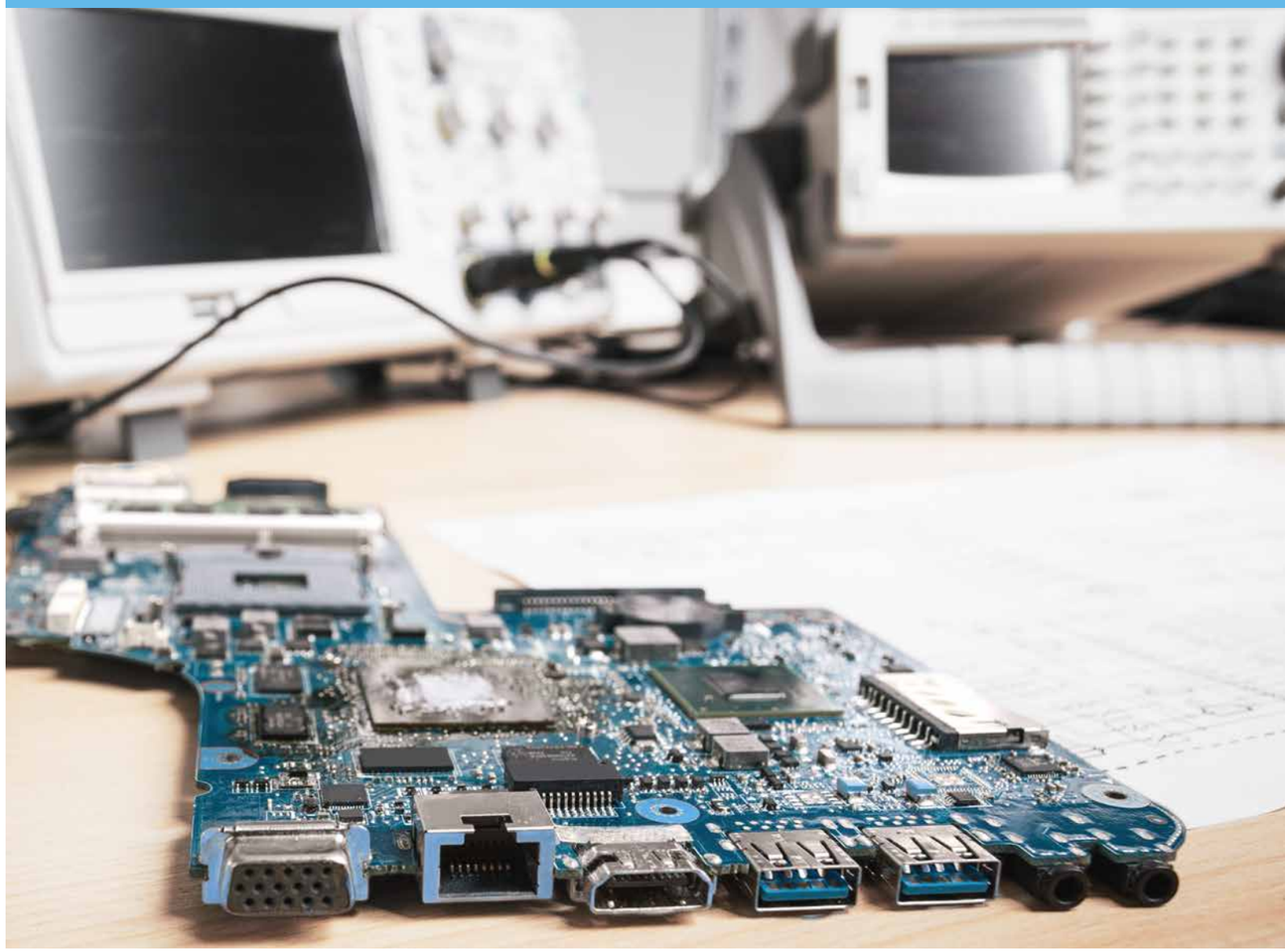
Optional: Integration of open source tests

There are several open source tools available to check the task jitter, the most common of them being "cyclictest"

When using Linux as Operating System, it is possible to run "cyclictest" in BE.'s Test Framework.

User Interface for cyclictest

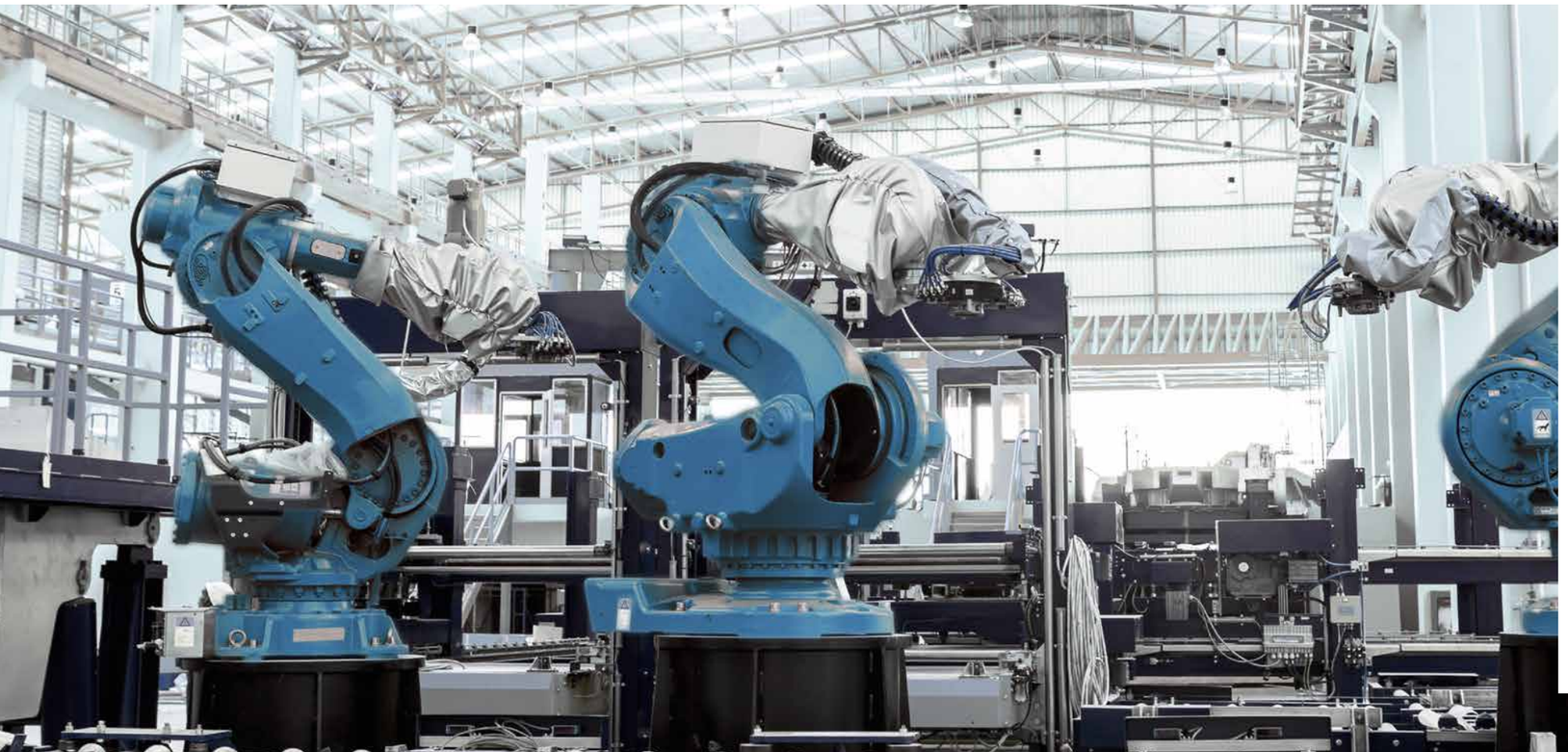
A complete set of test scripts are available to monitor the real-time performance of your system.



- Test parameters
  - Number of test cycles
  - Allowed maximum task jitter
  - Specific test arguments of cyclictest
- Test conditions
  - Hackbench (scheduler benchmark/stress test) can be started in parallel to cyclictest.
  - System specific Linux command. The user can implement scripts that create special ICS conditions.

INTUITIVE User Interface

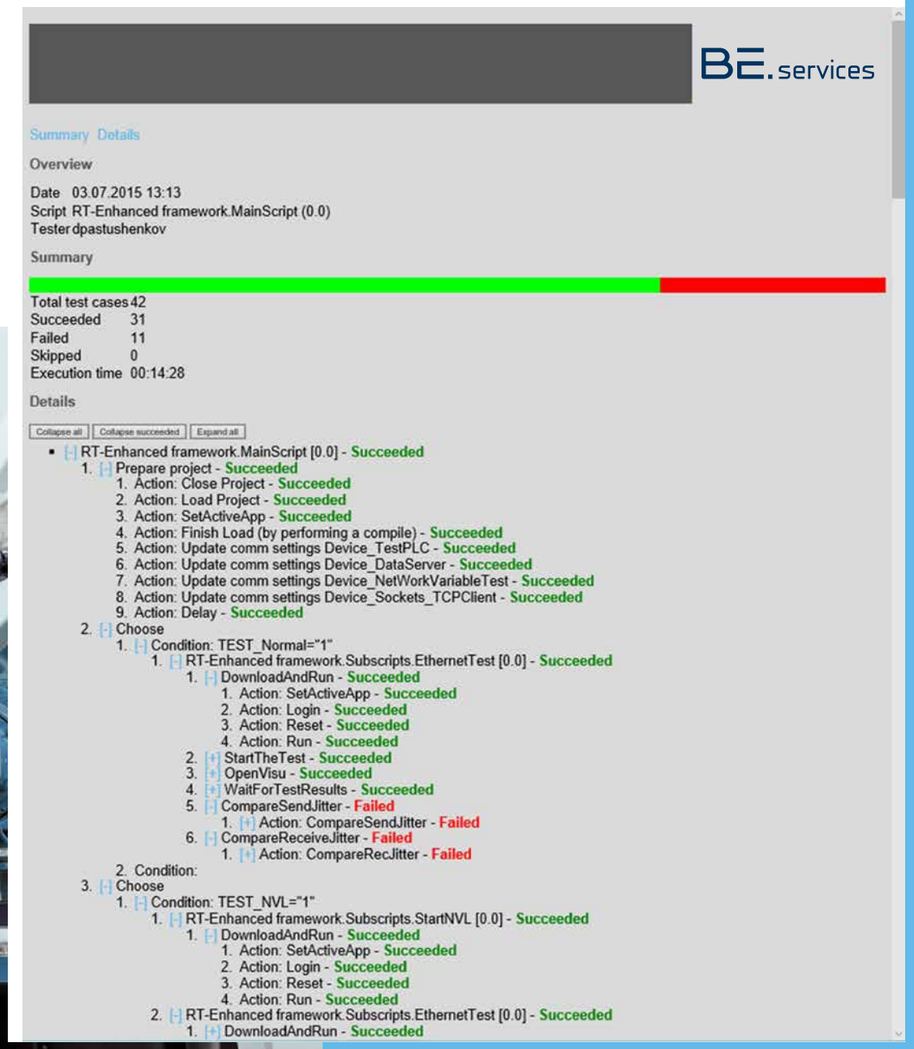
A comprehensive User Interface is available to run and monitor the project. Developed as CODESYS Visualization, it is embedded in the CODESYS project and eases the configuration and monitoring of the test execution.



Detailed test reports

If used in combination with the CODESYS Test Manager (product from 3S-Smart Software Solutions GmbH), the Test Framework delivers an extensive test report of the test process. Information like tester, test date and time, test conditions are displayed.

A detailed list of events with status is then provided as html format to review the errors during the test.



BE.services

Summary Details

Overview

Date 03.07.2015 13:13
Script RT-Enhanced framework.MainScript (0.0)
Tester dpastushenkov

Summary

Total test cases 42
Succeeded 31
Failed 11
Skipped 0
Execution time 00:14:28

Details

Collapse all | Collapse succeeded | Expand all

- [-] RT-Enhanced framework.MainScript [0.0] - Succeeded
  1. [-] Prepare project - Succeeded
     1. Action: Close Project - Succeeded
     2. Action: Load Project - Succeeded
     3. Action: SetActiveApp - Succeeded
     4. Action: Finish Load (by performing a compile) - Succeeded
     5. Action: Update comm settings Device_TestPLC - Succeeded
     6. Action: Update comm settings Device_DataServer - Succeeded
     7. Action: Update comm settings Device_NetWorkVariableTest - Succeeded
     8. Action: Update comm settings Device_Sockets_TCPClient - Succeeded
     9. Action: Delay - Succeeded
  2. [-] Choose
     1. [-] Condition: TEST_Normal="1"
        1. [-] RT-Enhanced framework.Subscripts.EthernetTest [0.0] - Succeeded
           1. [-] DownloadAndRun - Succeeded
              1. Action: SetActiveApp - Succeeded
              2. Action: Login - Succeeded
              3. Action: Reset - Succeeded
              4. Action: Run - Succeeded
           2. [+] StartTheTest - Succeeded
           3. [+] OpenVisu - Succeeded
           4. [+] WaitForTestResults - Succeeded
           5. [-] CompareSendJitter - Failed
              1. [+] Action: CompareSendJitter - Failed
           6. [-] CompareReceiveJitter - Failed
              1. [+] Action: CompareRecJitter - Failed
     2. Condition:
  3. [-] Choose
     1. [-] Condition: TEST_NVL="1"
        1. [-] RT-Enhanced framework.Subscripts.StartNVL [0.0] - Succeeded
           1. [-] DownloadAndRun - Succeeded
              1. Action: SetActiveApp - Succeeded
              2. Action: Login - Succeeded
              3. Action: Reset - Succeeded
              4. Action: Run - Succeeded
        2. [-] RT-Enhanced framework.Subscripts.EthernetTest [0.0] - Succeeded
           1. [+] DownloadAndRun - Succeeded

Delivery content

- Hardware:
  - TI-AM3359ICE2 (hard real-time network analyzer)
  - CODESYS Security Key
- Software:
  - CODESYS project including visualization
  - Test scripts
  - Firmware for TI-AM3359ICE2
  - CODESYS ControlWin SL

Additional services

If you prefer to send your controllers to BE.services, we also offer to execute the tests on your behalf in our lab and send you the test results.
We also provide consulting and development for the optimization of the real-time performance!

1 A standard Linux distribution was installed on the controller

Reference project: BE.services receives hardware from a client in order to test the real-time performance of their controller. To develop their business and extend the scope of applications, clients may target more demanding applications like motion control, robotics or, in a different industry, to add redundancy to the controller (redundancy requiring synchronization over Ethernet of both controllers, in real-time).

2 Controller with Real-time Preempt Patch

Results are not satisfactory for the targeted applications.

BE.services is asked to add real-time to the controller and the Preempt Patch from OSADL is integrated.

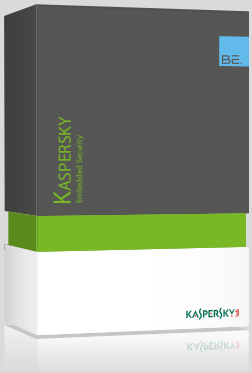
Results are much better but, in case of test on a longer period of time, we notice that the real-time is not stable, which would lead to malfunction.

BE.services provides consulting on optimization and develops a new Ethernet Driver as the jitter most probably is due to the standard Ethernet Driver.

3 Controller after BE.services optimization

The real-time performance was monitored in the different phases of the project and the client receives a new firmware with suitable real-time performance to cover new markets. The process is fully transparent as the client receives test results in all phases of the development.

Further products:



BE.services GmbH
An der Stadtmauer 4
87435 Kempten · Germany

Tel: +49-831-960-699-91
Fax: +49-831-960-699-90

www.be-services.com
info@be-services.com