

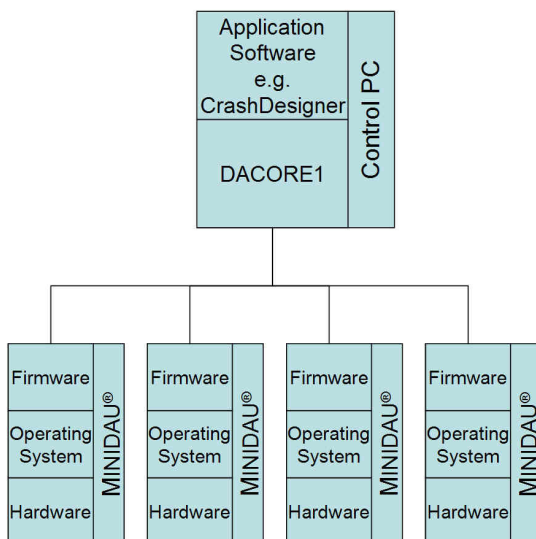
Support for Kayser-Threde MINIDAU® Systems by Third Party Software

Introduction

In the last 10 years Kayser-Threde has sold more than 35000 channels of our crash test data acquisition products. Most of these channels are operated together with the standard Kayser-Threde software packages, WinCarat and Autolab. And recently, Kayser-Threde has introduced its brand new WinCarat successor, CrashDesigner, which covers all aspects of crash test data acquisition from test preparation to test execution. However, Kayser-Threde is aware that there are special applications and certain customer requirements that make the use of a standard package challenging. In these situations our customers sometimes have the desire to use our MINIDAU® products together with software from other suppliers. The intention of this technical note is to explain the most feasible solutions for using Kayser-Threde hardware with third party software.

Software Structure

In order to find the best solution to the problem it is necessary to understand the structure of Kayser-Threde's software and the communication with the hardware. The following block diagram shows all components involved:



Hardware Levels

All the hardware functionality of a MINIDAU® (gain setting, bridge setting, shunt calibration, offset adjustment etc.) is programmable. The programming is done via control lines which are operated by a FPGA (Field Programmable Gate Array) on each amplifier board. The FPGAs on each amplifier board communicate via an operating system with the micro controller that is located on the controller module of each MINIDAU®. All the communication within each

MINIDAU® is not visible to the outside world and runs under full control of the real-time operating system. To allow an external controller (e.g. a standard PC) to access the functions of a MINIDAU® the device provides a higher level interface. This higher level interface consists of a command interpreter that converts ASCII commands into a format which is accepted by the operating system. We also call this interface the *firmware command interface*.

Software Levels

As mentioned before the firmware command interface is accessed via ASCII strings that use simple English words or abbreviations and usually a set of parameters. Of course, this interface is much simpler to use for a programmer than the internal communication. However, using application software which directly communicates with this interface is dangerous. Here are the main reasons:

- The firmware level interface is an internal interface of Kayser-Threde. Therefore, all of the documentation that is available is not designed or intended to be used by a third party software developer. In addition, a lot of this internal documentation is only available in the German language.
- The firmware level interface is subject to changes. The main reasons for changing this interface are either new functionality or the removal of bugs that might have an impact on the data integrity of the system. Due to the complexity of the functions inside a MINIDAU® Kayser-Threde cannot guarantee that all newer firmware versions are backward compatible to older versions. This means, the current firmware commands are different to the ones that were used four or five years ago. For a third party software developer this means that he has to change his application every time Kayser-Threde changes the firmware. In many cases he will only be aware of changes when the customer runs into problems after he or Kayser-Threde upgraded the system to the latest firmware. The situation for the customer is even worse: He has to decide between two scenarios:

Upgrade to the new firmware with the risk that the third party software does not work anymore

or

Stay with the old firmware without being able to take advantage of new features

- Firmware commands are device specific. This means, if several MINIDAU®s are used in a system (which is the case in 95 % of all

applications) the programmer must address each connected MINIDAU[®] individually. He also has to implement functions to determine the number of connected MINIDAU[®]s and has to add a layer that is able to handle a large number of devices.

- Using direct firmware commands within an application only makes sense for the hardware manufacturer himself since he can react to firmware changes immediately.

Because of these reasons Kayser-Threde strongly suggests not to use the firmware level interface to integrate MINIDAU[®]s into third party software. In the end it will be the customer who suffers from such an implementation!

Several years ago Kayser-Threde decided to offer a much better software interface for the developers of third party software. This API (Application Programming Interface) is called DACORE1 and provides a high level interface for Kayser-Threde hardware on a system level (remember that the firmware is on the device level!). System level means that the API takes care of the correct programming sequence of several MINIDAU[®]s and Airbag Timers including important functions such as trigger settings etc.

Example: *To perform an offset adjustment on 5 connected MINIDAU[®] devices it will be necessary for the firmware level programmer to send several lines of ASCII commands to each connected MINIDAU[®]. He also has to implement functions that handle the response of each MINIDAU[®]. The DACORE1 programmer basically uses one command in his application to perform the same programming sequence while DACORE1 takes care of all necessary programming steps and handling of hardware responses. This reduces the time to integrate Kayser-Threde hardware into a third party software dramatically and in addition, it leaves the low level communication and error handling where it should be: In the hands of the hardware manufacturer.*

DACORE1 runs on all popular Windows platforms and provides support for MINIDAU[®]s, Stationary MINIDAU[®]s, nxt (in MINIDAU[®] emulation mode) and 8-channel Airbag Timers. The API is well documented and KT guarantees that in case of firmware changes a new version of DACORE1 will be available immediately that incorporates all firmware changes and still provides a consistent interface to the application. This means, it is not necessary to make

changes to the third party software package in case of a firmware change. It is as simple as providing the customer with a revised DACORE1 and the system will remain stable.

Today, companies like FALCON, Messring, and MICROSYS are successfully using DACORE1 to include support for Kayser-Threde hardware into their software packages. In return, Messring makes an API available to Kayser-Threde that allows the integration of Messring hardware in Kayser-Threde software.

Conclusion

It is ideal for manufacturers of data acquisition hardware to provide support for their systems so that software from other suppliers can be utilized. In this context it is extremely important for the customer that the support within the software is done via a high level API that is fully supported and documented by the hardware manufacturer and not through native firmware commands which may be revised or updated at any time. Furthermore, customers should be cautious of software that uses firmware level commands if it does not come from the manufacturer of the hardware directly.

On the other hand for the hardware manufacturer it is important to have the freedom to make changes in the firmware to improve the functionality and integrity of his hardware and at the same time only maintain one interface to third party developers.

Customers should be also extremely suspicious of manufacturers who claim to have hardware that is compatible with MINIDAU[®] systems on firmware level. Because of the reasons explained above it is extremely unlikely, that such an emulation, especially if it was developed without the support and approval of Kayser-Threde will ever work reliable enough to be used in an expensive application such as crash testing.

By providing APIs to each other and to third parties, Messring and Kayser-Threde have shown that they highly respect the customer desires by providing the best technical solution.