CAN-based safety parameterization

The development and maintenance cost of safety related software packages is much higher than for non-safety functions. Using parameterization, software components can be standardized across multiple machine variants.

Using a suitable method, the machine designer can adapt software components to the particular characteristics of a machine configuration without modifying safety relevant software to minimize the certification and maintenance effort for safety relevant modules. Parameterized programming is often mentioned as a technique for a reliable reuse of software. In this technique, modules are parameterized over very general interfaces that describe required properties of an environment for the module to work correctly. The reusability of the software may cut down costs of the rising demands on the flexibility and reliability of software. “The basic idea of parameterized programming is to maximize program reuse by storing programs in the most general form possible. One can construct a new program module from an old one just by instantiating the relevant parameters.”

Use-cases of a parameterization solution for mobile systems

Figure 1 shows functional use-cases of a mobile system from different perspectives. All use-cases need operations that involve “safety parameterization” in the system. The application that uses the parameters runs in the electronic control unit (ECU), which can be seen as the brain of the mobile working machine. A solution must allow the ECU application developer to define parameter sets during the development phase of the ECU application. It must also enable the operator to read and write individual parameter values from a personal computer (PC) tool or by using a non PC-based human-machine-interface (HMI). The ECU application should use parameter values stored in the ECU’s memory, too.

Available standardized parameterization tools do not fulfill the requirement of writing individual parameter values in the field, which is a special use-case for mobile applications. For instance, the mobile machine needs parameterization during the initial operation for adjusting sensors and actuators. In addition, it might be necessary to exchange components during a preventive maintenance. The PC toolchain Kefex is used as an example to demonstrate different tasks during the development and maintenance life cycle of an ECU application. Its first integral part is the tool RAM-View, which supports the use-cases for parameterization.

Software-based safety parameterization

The component RAM-View is used to read and write parameters and diagnostic variables from or to the ECU’s memory and supports configuration and monitoring. It can be used to create and edit parameter sets and to view or modify the variables on the ECU, too (see Figure 2). The Kefex client, which also runs on a PC, provides components for communication with the ECU and displays customer defined values on the HMI. Different dynamic link libraries (DLL) support CAN interfaces such as from Peak, Vector, Ixxat, and CPC-PP to establish the communication between the PC and the ECU.

To work with variables defined in RAM-View, it is necessary to run the Kefex server on the ECU side. The server provides mechanisms to read and write memory contents from and to a PC tool and reads parameter values from the ECU hardware abstraction layer (ECU HAL). The server supports working with multiple projects on one ECU, too. These multiple RAM-View projects can be linked with one application. Now, how can RAM-View become a tool for safety parameterization of safety relevant software modules developed in either the safety variant of the programming language “C” or IEC-61131 (Codesys Safety SIL-2)?

Safety parameterization techniques

Both safety relevant standards EN ISO 13849 and EN 62061 describe the same requirements towards software-based safety parameterization. RAM-View fulfills these requirements. As a supplier of 32-bit safety ECUs like the ESX-3XL or ESX-3XM, Sensor-Technik Wiedemann (STW) provides the dedicated software tool RAM-View Safety for safety parameterization. It delivers appropriate actions to verify the tool configuration and prevent unauthorized modification with password features. The required measures to control valid values are assured by the cooperation between the Kefex client and server.

A suitable action to handle the data corruption of single parameters before transmission is reading the values and confirming their validity through the operator. To control the effects of errors arising from the parameter transmission...
process, incomplete parameter transmission, and the effects of faults and failures of the hard- and software, the Kefex server verifies checksums. Kefex RAM-View Safety fulfills all requirements of software-based safety parameterization realized by a suitable special procedure. This includes the confirmation of input parameters to the safety ECU by retransmission of the parameters to the parameterization tool. The procedure also includes a confirmation by an operator and an automatic check. Diverse functions avoid systematic failures. These functions cover encoding/decoding within the transmission/retransmission process and visualization of the non-safety and safety-related values to the operator.

Interactions between RAM-View and ECU

The system is designed in such a way that safety critical decisions are either made by a safety ECU software or are covered by a clearly defined parameterization process. PCs cannot make any safety related decisions. Therefore, although the PC calculates all checksums, the decision about the correctness of these checksums is only made by the safety ECU. The variable description tables exported by RAM-View are linked with the ECU application. RAM-View therefore is a “T3” off-line support tool as defined in DIN EN 61508-4. This standard was applied to the tool qualification process. As a further defensive measure, the Kefex server performs consistency checks on the exported data.

The whole communication between the PC and the ECU is considered a “black channel”. It includes writing the payload data to the ECU’s memory. The server ensures that accidental accesses from RAM-View or other PC tools do not affect safety critical parameters. This is achieved by using the memory protection mechanisms implemented within the ESX-3XL and ESX-3XM. By utilizing the mechanism, the recognition of an accidental manipulation of safety data can be ensured and the PC tool does not need to be considered as an online support tool according to DIN EN 61508.

Process for creating and writing pre-defined safety parameter sets

The following process is used for creating and writing safe parameter set files. It can be used to pre-define parameter values for deployment to a number of ECUs, e.g. end-of-line programming in series production. The following steps are performed for creating parameter set files on the ECU:
1. configuring desired parameter values,
2. reading and storing values of selected parameter lists including their checksums,
3. checking the created parameter set file.
When writing parameter set files to the ECU, the following steps are performed:
The PC tool
1. checks the checksums of the parameter values and fails if they do not match,
2. writes the contents of the parameter values to the ECU’s memory,
3. and sends checksums of the new data.
The ECU then checks whether the received checksum matches the received data and addresses. An operator reading back the values and confirming their correctness.
This approach leads to a clear separation between safety-relevant and non-safety-relevant parameters in:

- The project part on a personal computer: the project with the safety-relevant parameters does not need to be touched when one is only changing non-safety relevant parameters,
- The ECU volatile memory at run-time, which utilizes the memory protection mechanism,
- The ECU code, which entails separate variable description tables; it is not necessary to change the file with safety-relevant parameters if only the non-safety relevant parameters are changed.

As a result, the user will see a reduced testing effort as long as only non-safety relevant parameter sets are changed.

Conclusion

Kefex RAM-View Safety fulfills the standards of EN ISO 13849-1 and DIN EN 62061 for safety based parameterizations. The solution is designed in such a way that all safety critical decisions are either made by the safety-related ECU system or are covered by its well-defined interaction process between the PC tool and the safety ECU. As a result, the use-cases shown in Figure 1 can be covered. RAM-View provides functionality for the definition of parameter sets during the development phase of the ECU application. The processes for creating and writing the parameter set files prevent undesired changes in data by storing the data as a black box on the PC side. If any part of the data is changed on the way, the ECU detects this by controlling the checksums.

The operator can use a process for parameter modification, which ensures that no undesired changes occur in the data during the transmission between PC and ECU memory. The transmission and presentation paths are diverse and allow the detection of systematic problems. The ECU application uses parameter values stored in the ECU's memory. A multiple instantiation supported by RAM-View achieves a clear separation between parameters that are used for safety critical and non-interfering functionality. The result is a reduced testing effort as long as only non-safety relevant parameter sets are changed. Kefax RAM-View is provided to be used in “C” applications and is designed to be integrated into a Codesys Safety SIL-2 run time system.

Process for parameter modification

The following process is used to change one or more parameters when there is no pre-defined parameter value file. This could happen during the modification by a service technician in the field who needs to set individual parameter values from a PC or HMI. The PC tool or HMI

1. writes the parameter values to the ECU,
2. reads back the parameter values from the ECU,
3. reports read back values to the application layer HMI,
4. waits for approval by the operator,
5. calculates the checksums,
6. and sends checksums of the new data values.

The ECU then checks whether the checksums received match the received data and addresses. This procedure prevents undesired changes in the data during the transmission between the PC and ECU memory. The transmission and presentation paths are diverse and allow detection of systematic failures. The parameter values are checked to be in a valid range and hardware faults in the memory are detected. Of course, if any part of the data was changed on the way, an incorrect checksum will identify this.

Independence by multiple instantiation

A clear separation between parameters that are used for safety critical functionality and parameters that are used for non-interfering functionality is achieved by the Kefex server. It allows multiple instantiations with separate sets of parameter definitions.

Two (or more) different RAM-View projects, each containing a parameter set definition, can be defined with parameters used for safety or non-safety functionality. This

Figure 2: Relevant components for safety parameterization

is not required as the consistency of the data stored in the file is ensured by checksums.

Author

Kai Niestroj
Sensor-Technik Wiedemann
www.sensor-technik.de
info@sensor-technik.de
Powerful Control Units for High-Safety Applications: HY-TTC 500 Family

Flexibility & Usability
- Single controller for whole vehicle for centralized architectures
- Extensive I/O set with multiple software configuration options per pin
- Open programming environments C, CODESYS® V3.x and CODESYS® V3.x Safety SIL 2

Safety
- TÜV-certified according to IEC 61508 (SIL 2) and EN ISO 13849 (PL d)
- ISO 25119 AgPL d certifiable
- CODESYS® Safety SIL 2 including support for CANopen® Safety Master and easy separation of safe / non-safe code
- Safety mechanisms in hardware to minimize CPU load
- Up to 3 output groups for selective shut-off in case of safety relevant fault
- Safety companion and safety mechanism in hardware

Connectivity
- Up to 7 CAN interfaces
- Automatic baudrate detection and configurable termination for CAN
- Ethernet for fast download and debugging purpose

Performance
- 32 bit / 180 MHz TI TMS570 dual core lockstep processor (ARM architecture)
- Up to 2.3 MB RAM / 11 MB Flash
- Floating-point-unit

Robustness
- Automotive style housing suited for very rough operating conditions
- Total current up to 60 A

www.ttcontrol.com/HY-TTC-500-Family