DOI: 10.2195/lj_Proc_holzweissig_en_201610_01
URN: urn:nbn:de:0009-14-44720

HiPi – A concept for a compact, IoT-enabled safety controller

HiPi – Ein Konzept für eine kompakte, IoT-fähige Sicherheitssteuerung

Peter Holzweissig*
Tommi Kivelä*
Silvia Vélez León*
Markus Golder

Sichere Mechatronische Systeme der Intralogistik (SIMESI)
Institut für Fördertechnik und Logistiksysteme (IFL)
Karlsruher Institut für Technologie (KIT)

** Equal contribution*

Intralogistic applications require ever more degrees of freedom for the realization of complex processes. Use of modern technologies in the control of these applications enables human-machine cooperation and interaction. The requirements to the safety of these applications increase together with the complexity. A concept for a new safety controller targeting these applications is illustrated in this work. The compact and IoT-enabled concept controller will be suitable for the implementation of safety functions especially for material handling devices.

[Keywords: intralogistics applications, safety controller, Internet of Things, security]

Intralogistische Anwendungen benötigen immer mehr Freiheitsgrade zur Ausübung komplexer Vorgänge. Der Einsatz moderner Technologien zur Steuerung solcher Anwendungen ermöglicht die Zusammenarbeit und die Interaktion zwischen Mensch und Maschine. Die Anforderungen an die Sicherheit dieser Anwendungen steigen mit Zunahme der Komplexität. In dieser Arbeit wird das Konzept einer neuen Sicherheitssteuerung für den Einsatz in solchen Anwendungen dargestellt. Die kompakte und IoT-fähige Steuerung eignet sich für die Implementierung von Sicherheitsfunktionen, besonders für Fördermittel der Intralogistik.

[Schlüsselwörter: Intralogistikanwendungen, Sicherheitssteuerung, Internet der Dinge, Security]

1 INTRODUCTION

Intralogistics systems are subject to the Machinery Directive 2006/42/EG [EU06]. The easiest way to fulfill the requirements of the Machinery Directive is the utilization of a type C product standard conforming to it, which also describes the emanating risk. If a type C standard is not available, the machine producer has to use a generic type B standard and perform the risk assessment for the overall system. To fulfill the safety requirements for such applications, generic safety control systems are used. For the development of an electronic system, the standards DIN EN ISO 13849-1 [DIN16a] or DIN EN 62061 [DIN16b] give the parameters to the functional safety in safety levels, the Performance Level (PL) and the Safety-Integrity Level (SIL), and how to reach them [DIN16a, DIN16b]. For intralogistic systems, the required PL or SIL is normally not as high as in other factory automation sectors, often a safety level of PL c or PL d, or SIL 2 is sufficient.

In this work, a concept for a compact safety controller, targeting especially intralogistic applications, is presented. In chapter 1, the state of the art in safety control systems is presented and motivation for the concept is given. Chapter 2 presents the concept device hardware, software and communication concepts in more detail. In chapter 3, potential target applications for the device are presented as a further motivation and to provide a basis for the requirements for such a device for further research work. Chapter 4 concludes the work and in chapter 5 future research work is discussed.

1.1 STATE OF THE ART IN SAFETY CONTROL SYSTEMS

Safety control systems can be roughly divided into safety devices and safety controllers, where safety controllers are also available as integrated systems with a non-safe part.

Safety devices have a fixed Boolean logic which can sometimes be parametrized over switches. They are often rated based on the price and required space per digital IO point. To reach a more complex Boolean logic a combination of multiple safety devices is necessary, which need to be hard-wired. The infrastructure of systems built with safety devices represent in most cases a central system architecture.

On the other side there are safety controllers, which are available as compact or modular systems, where the logic can be programmed according to [DIN14b]. The system can also be decentralized using remote IO components, which are connected over safety fieldbus systems. Due to their range of functions these systems are quite expensive.

Both types of safety systems are usually designed to run inside an electric cabinet, where they are rail-mounted and supplied by an external 24V power supply. Moreover, they are normally engineered with a “top-down-approach”, which means that they are designed to fulfill the highest safety requirements according to [DIN16a] and [DIN16b] and can also be used for applications with lower requirements. The development and certification of such systems is very time- and cost-intensive.

1.2 Motivation

According to [Baua12], half of the fatal accidents involving work equipment occurring in the period from 2001 to 2010 happened in the material handling devices field, especially with vehicles, cranes and forklift trucks, as shown in Figure 1. Therefore, there is an increasing need in automating the monitoring and controlling of the applications, in which such devices are being used. Thus, the reliability of these systems can be increased and the risk of human failure causing a fatal accident is greatly reduced.

Moreover, with the Industry 4.0 revolution [BMWi16], where as many tasks as possible are being realized by machines but still in cooperation with humans, poses great challenges for safety controllers. The safety of the workers depend greatly on the safety controllers of the machines. With the increased interconnection between machines and of the machines to the Internet, security is also of paramount importance to safety controllers in the future.

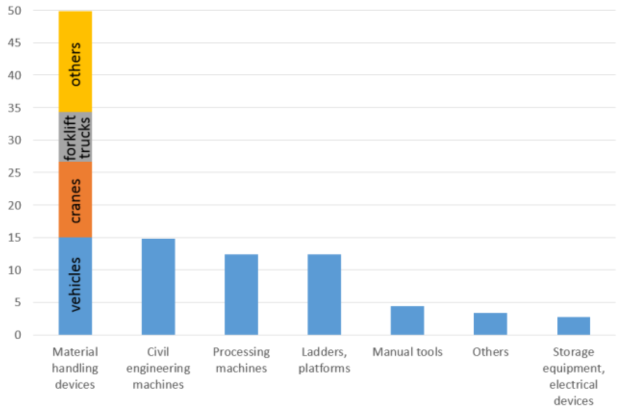

Figure 1. Overview on amount of accidents with different work equipment between 2001-2010 [Baua12]

Since modern intralogistics systems, such as automated guided vehicles (AGV), require a high flexibility in usage and have only limited space for safety systems, neither of the presented safety control systems are a good fit for these kinds of applications. To reach that flexibility for autonomous systems, the human-machine-interface (HMI) gets much more important, as shown in research projects like [Tre13]. For human-machine cooperation to be viable in the near future, the barricade between them has to disappear. Thus the usage of intelligent safety control systems is essential.

The pursued approach here is to create a non-oversized safety controller, which satisfies the safety requirements of most of the intralogistic applications (“bottom-up-approach”). Simultaneously, the controller will also be able to provide the interconnectivity and newer technologies required for the Internet of Things (IoT), which are usually not covered by safety control systems on the market. The concept, as derived from this basis, is discussed next.

2 Concept

In this section, the basic ideas of the HiPi concept and how the hardware and software designs could look like will be explained. A main principle of the design is the separation of the safety-related parts of the device from the non-safety-related parts. This principle is carried through the hardware to the software and the communication concept. The principle provides flexibility for the whole device, while avoiding the need for re-certification of the safety functions if changes to non-safety-related sections are required.

The safety-related and the non-safety-related parts need to be able to communicate with each other, however, to be able to provide functionality such as for example the displaying of operational information or the possibility to securely set the parameters of the safety functionality over a web-interface running on the non-safety-related section. For this purpose, a communication interface will be designed and certified alongside the safety-related section of the device.

2.1 Hardware

The hardware on the HiPi will be separated into safety-related and non-safety-related parts. Safety-related functionality will be running on an individual multicore System-on-Chip (SoC) and the non-safe functionality on a separate SoC. The SoCs will be physically separated on to their own respective printed circuit boards (PCBs). The PCBs with the SoCs will be the core of the device, but an additional PCB is needed for interface-electronics and a power supply. The hardware architecture of the concept is shown in Figure 2.

The non-safe PCB will be a custom board inspired by popular single board computers such as the Raspberry Pi. The main chip on the non-safe PCB can be any commercial SoC, which is capable of running a variant of the Linux-operating system (OS). The goal is to be able to take advantage of the ecosystem around the Linux-running single-

DOI: 10.2195/lj_Proc_holzweissig_en_201610_01
URN: urn:nbn:de:0009-14-44720

board computers, while creating custom hardware, suitable to be used in an industrial environment. The non-safe circuit board should include interfaces for internet-communication through LAN and WLAN. Additionally, the hardware should provide interfaces for implementing non-safe functionality, such as connecting cameras or other sensors to the device using for example RS485, SPI or USB-interfaces.

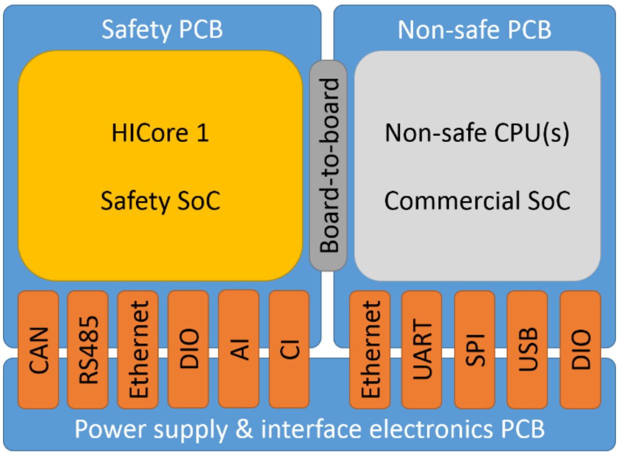

Figure 2. HiPi hardware architecture block diagram

Multiple safety certified microcontrollers or SoCs are available on the market, where a certified safety OS and firmware are supplied in addition to certified hardware. Different suppliers include for example Renesas Electronics Corporation [Renesas16], NXP Semiconductors [NXP12] or Texas Instruments [TI16]. The safety concept as discussed here will be based on the HICore 1 safety SoC [Hay14], which consists of two safety central processing units (CPUs) in a fully redundant 1oo2D architecture and a communication CPU providing black channel communication inside a single chip. The name of the HiPi-concept originates from the idea of combining the HICore 1 with a Raspberry Pi. The HICore 1 includes a large amount of the required functionality for safety applications on-the-chip. However, additional interfacing electronics still need to be provided around the chip. These will be included on the safety PCB.

The safety SoC and surrounding hardware will remain fixed throughout the product lifetime, as the components are supplied for industrial use and thus have guaranteed long product supply cycles, and will be available long in the future. The safety certification will only be done to the safety PCB and the interfacing electronics. The non-safe PCB will include commercial components, which will not have as long supply cycles. The concept is such, that this commercial hardware can be updated to newer components in the future, while not affecting the certification of the safety hardware, since that and the interfacing electronics will remain fixed.

The safety PCB interfaces will include a dual CAN-interface, Ethernet-interface for flashing the safety CPU and a standard RS485-interface for connecting additional sensors, such as RFID systems. Since HICore 1 doesn’t have analog inputs (AI), the available SPI-interface is used to connect an analog-to-digital converter (ADC) chip for realization of the required AIs. In addition, the safety PCB will include digital inputs and outputs (DIO), which can be utilized for digital +24V signals, or for implementing relay outputs on the interface electronics board. HICore 1 also has counter inputs (CI) which will be used to connect e.g. incremental rotary encoders in the variants up to three quadrature counters. Safe fieldbus communication can be implemented over a single CAN-connection using CANopen Safety [DIN16c]. Other possibility for achieving redundant communication will be by using both CAN-channels physically separated. The non-safe and the safety-PCBs will be able to communicate to each other over a fixed board-to-board interface, which will be created and certified over the device creation. The communication concept is discussed in more detail in section 2.3.

The HICore 1 is a cost-efficient and compact solution for limited size safety applications, but it also poses a limitation to the capabilities of the safety functions the HiPi will be able to perform. The safety CPUs are based on an 8-bit architecture with a clock-frequency up to 135MHz [Hay14]. The safety applications will be run as single threads on both the safety CPUs on top of the safety OS and firmware. This limits the complexity and reaction times possible to achieve with the device, but the authors are confident that it will be a suitable and flexible platform for most required safety functions in the target application field of material handling devices.

2.2 SOFTWARE

The HiPi will not be a programmable logic controller (PLC) as such, since the application implementation will be done using programming languages such as embedded C++ [Pla97] on the safety CPUs, and C/C++ and Python on the non-safe CPUs instead of the standard PLC-programming languages according to [DIN14b].

In terms of software, both used platforms are considered independent and running non-reactive to each other. This means, that a safety application will be running on the safety CPUs in parallel to a standard program on the non-safe CPU and if something happens to the non-safe CPU the safety functions will still be executed (see also Figure 2).

Figure 3 presents all three programmable units of the HiPi: the safety CPUs running the safety-related application program, the communication CPU and the commercial SoC running Linux-OS-based software. The communication CPU program will be fixed and allows the communication over all the planned interfaces. The configuration of the interfaces can be done over the safety application program. Different safety functions will be available as template programs for different types of applications, where the controller is planned to be utilized (see chapter 3). The different applications can then be easily downloaded via the programming interface to the safety CPU. Therefore, the controller can be implemented on demand, according to the Plug & Play concept, providing high flexibility to meet the current industry requirements. The safe software is architecture-dependent, since the several safety SoC suppliers do not use the same processors for their systems and the underlying OS and firmware will also be supplier-dependent.

On the other hand, the non-safety-related software will be implemented in a high-level programming language, such as Python, on top of the Linux-based OS, separating it from the underlying hardware. Thus, it can be easily transferred to upgraded hardware and OS in the future with reduced workload. Therefore, the availability problem related to consumer electronics, which are not guaranteed to be in the market as long as industrial products, is solved. Moreover, a significant part of the non-safety-related software will not be application specific, which will allow software re-use between different target applications.

While the access to the IoT is provided via the non-safe platform, security measures are required in order to prevent the influence of external parameters on the safe application. Thus, the information flow between the two platforms – especially from the non-safe to the safe part – is monitored, assuring that the transferred data is correct and reliable. The communication concept is introduced in more detail in the next section.

2.3 Communication Concept

The HiPi system is characterized by an internally separated PCB design. To implement a board-to-board interface between the safety PCB (HICore 1) and non-safe PCB (commercial SoC), which enables communication between the safety- and non-safety-related parts while maintaining separation and independent operation, the architecture shown in Figure 3 will be used.

The HICore 1 consists of a safety processor and a communication processor, which are linked by a Dual-Ported-Ram (DPRAM). The communication processor contains several communication interfaces, for which the functionality can be programmed according to application need, which are: an Ethernet interface, two UART based interfaces (RS232 and/or RS485), a SPI and two CAN interfaces, where the CAN interfaces are not in focus for the board-to-board communication. Ethernet is a modern technology but has an oversized frame for the amount of exchanged information and is security sensitive. The UART allows a simple asynchronous serial communication with a small communication frame. The SPI is very fast and is based on a Master-Slave communication. The HICore 1 firmware includes SPI master functionality. Commercial SoCs with Linux OS also typically provide SPI functionality as an SPI master, but not as an SPI slave. Thus, it would be necessary to implement SPI slave functionality for one of the systems. Therefore, UART is the most attractive interface for the board-to-board communication.

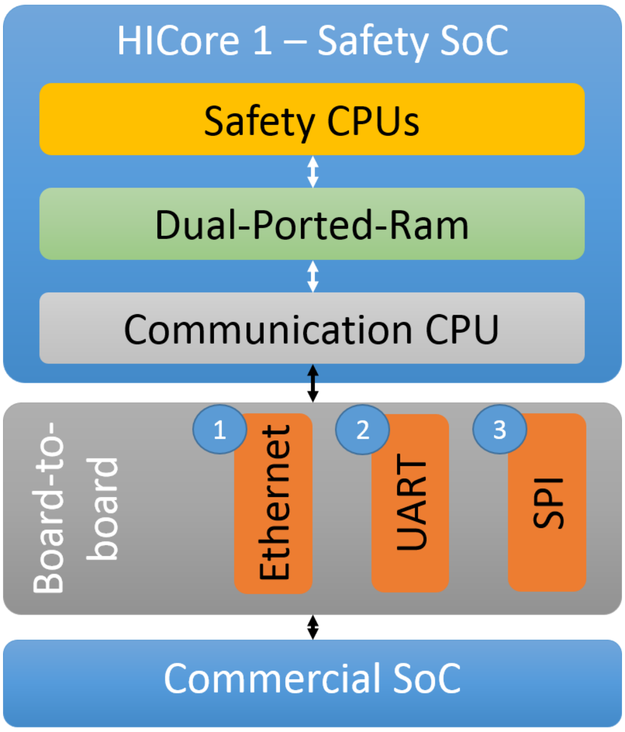

Figure 3. Board-to-board communication architecture with possible implementations

The next functionality is safe communication between multiple HiPi devices, like introduced in the subsection “Hardware”, is visualized in Figure 4 and Figure 5. The two CAN interfaces from the HICore 1 will be used directly. A CANopen Safety communication stack will be implemented over them. The validation of the safety information will take place inside the safety CPUs.

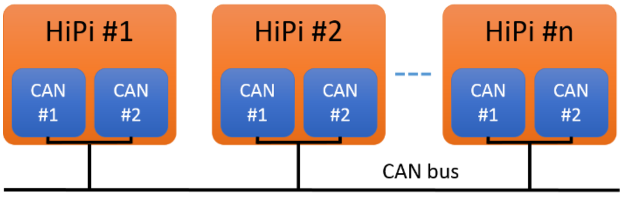

Figure 4. Safe communication network with a single CAN bus

The redundant CAN bus, as shown in Figure 5, can be used to increase the availability of the HiPi network. It can also be used to separate multiple single CAN bus networks (non-safe). Due to the performance restrictions of the HICore 1, especially due to the amount of available memory, the number of CANopen Safety participants cannot be as high as the limit of 63 given by the standard [DIN16c].

The last important feature of the HiPi is security in IoT context. The Linux based commercial SoC has ideal conditions to realize a secure gateway for the system. One

DOI: 10.2195/lj_Proc_holzweissig_en_201610_01
URN: urn:nbn:de:0009-14-44720

planned feature which is currently not state of the art in safety controllers is the possibility to access the system via a unique IPv6 address. Another security feature is to define algorithm based user access, so that the login information is changing continuously. These features are normally only possible over an external router or gateway and now directly available over the safety controller.

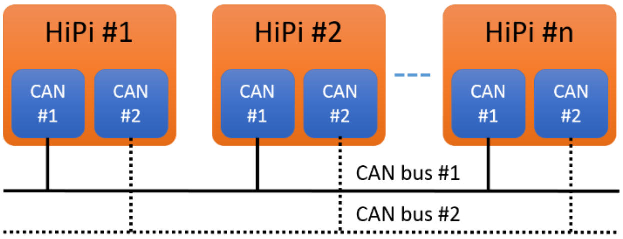

Figure 5. Safe communication network with a redundant CAN bus

3 APPLICATIONS

Typical applications, where the HiPi could be utilized, will be discussed next. The device will be suitable for, for example, supervising the safety of the motion along a single axis. Therefore, to monitor multiple axes, multiple devices would be used. These devices could then implement larger-scale safety functions decentrally by communicating with one-another through a safe communication medium. Main example here will be possible safety functions in crane applications. Other possible applications from the field of material handling devices are also discussed.

3.1 SAFETY FUNCTIONS IN CRANE APPLICATIONS

Functional safety requirements for crane applications are defined by a combination of several type C standards, such as [DIN10], [DIN13] and [DIN14a]. The combined requirements from these standards can be derived for the safety-related parts of the control system, for devices which are electronic or programmable, as follows:

- Standard control systems: PL c and at least category 2.
- High-risk applications: PL d and at least category 3.

The resulting device is to be suitable for both standard and high-risk applications, thus it needs to be able to reach PL d and at least category 3.

According to [DIN13], the control system of a crane shall implement the following safe functions, unless they are eliminated by other means:

- Overload protection
- Limiting of relevant motions
- Emergency stop
- Overspeed control for variable speed hoisting drives

HiPi will be a suitable device for implementing each of these functions for a specific movement. Overload and overspeed protection are discussed further as examples in the following sections. Additionally, possibilities to implement other safety functions by co-operation of multiple units is discussed shortly. The applications are discussed in this section with the focus on the functional safety part. For all the crane applications, the non-safety-related part of the HiPi can be utilized for different functions, such as providing a web-based interface to the device, perform non-safety related control functions or provide data logging of the operation of the crane or hoist.

Especially in small hoists, the space available in the electrical cabinet is often very limited. Small size is a strict requirement for safety controllers in these applications, rendering the HiPi to be especially suitable.

3.1.1 HOIST OVERLOAD PROTECTION

Overload protection is required for all size hoists. In smaller hoists, dual-speed induction motors with contactor controls are still the most typical solution. This is due to a combination of low dynamic requirements and user preference for simple operation and high robustness and reliability. In hoists for heavier loads or for applications where better dynamic performance is required, frequency converters are used.

The implementation of the overload protection depends on the hoist control concept. HiPi will be suitable for implementing the overload protection for a contactor controlled hoist drive with safe digital I/O or for a frequency converter driven hoist there will also be a possibility to implement the function using safe fieldbus communication between HiPi and the frequency converter.

An example architecture for overload protection, fulfilling PL d and category 3 is illustrated in Figure 6. HiPi receives the hoist control signals, from a source such as a pendant or a radio controller, as digital input signals. In addition the main contactor control signal is supplied through the device. The main contactor is responsible for supplying the mains voltage forward to the motor contactors, which further also supply the brake contactors. The hoist load is supplied to the HiPi using redundant load sensors, which are typically strain gauge-based, but other sensor types can be used. The outputs from the device are the control signals forward to the hoist and also the control signal for the main contactor. If the hoist load is higher than a previously set limit, running upwards is no longer permitted.

Internally the HiPi processes the input signals using a dual-channel architecture, where the control and contactor inputs, as well as the separate load sensor inputs, are processed individually by both CPUs. To fulfill category 3, the

DOI: 10.2195/lj_Proc_holzweissig_en_201610_01
URN: urn:nbn:de:0009-14-44720

CPUs need to supervise each other. The redundancy in the actuators is achieved by controlling and receiving feedback from both the main contactor and the hoist control contactors supplied by it.

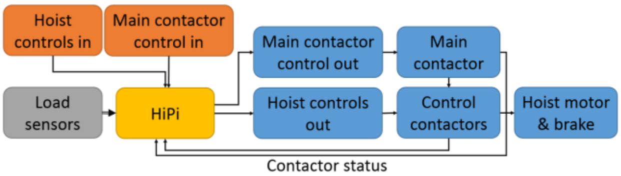

Figure 6. *Architecture for hoist overload protection: Hoist with contactor control.*

An example list of required interfaces for this application architecture would be:

- Load sensors: 2 differential analog inputs
- Hoist controls: 3 digital inputs, hoist up & down, fast speed activation
- Main contactor control in: 1 digital input
- Hoist & main contactor control out: 4 digital outputs
- Feedback for contactor status: 2-5 digital inputs. Depending on the wanted functionality, the main contactor and the motor contactors' feedback can be put in series and supplied to a single input. In the case of a dual brake contactor, these can likely be serialized as well.

Additionally other IO will likely be wanted for additional functionality, but listed here are the required IO points for the core function.

An example architecture for hoist overload protection for a frequency converter driven hoist is illustrated in Figure 7. The difference to the contactor controlled hoist is that the frequency converter, supplied through the main contactor, is now responsible for controlling the hoist motor and the brake. The communication between the HiPi and the frequency converter in the figure can be implemented using safe communication over a fieldbus, such as CANopen Safety [DIN16c], as illustrated in the figure. Safety bus is not strictly necessary, the implementation would also be possible with traditional digital IO-control of the frequency converter, given that the frequency converter has a safe certified inputs for stopping the motor and closing the brake. Similarly to the architecture in Figure 6, the redundancy in the actuator-side is achieved by controlling the main contactor and the frequency converter through the HiPi. Internal processing of the signals on the HiPi in this architecture and the previous one are similar.

The hoist overload application, regardless of the control architecture of the hoist, requires a moderately fast reaction time. The cost-efficient solution for the load sensors are still circuits, which provide an analog signal, which for safety applications will typically be a 4-20mA current signal.

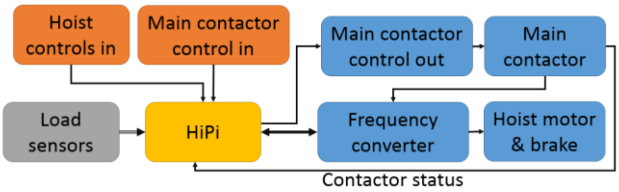

Figure 7. *Architecture for hoist overload protection: Hoist with frequency converter control. The bolded line represents safe communication over fieldbus.*

The required IO is very similar for this architecture, with the difference of reduced IO-points required for control outputs and feedbacks if a fieldbus is utilized in between the frequency converter and the HiPi.

3.1.2 OVERSPEED PROTECTION

Overspeed protection is required for frequency converter driven hoists. A possible architecture for overspeed protection is illustrated in Figure 8. To achieve PL d with Category 3, either two redundant regular speed sensors or a single PL d, category 3 certified safety speed sensor. The detected speed can also be supplied to the frequency converter for speed control purposes.

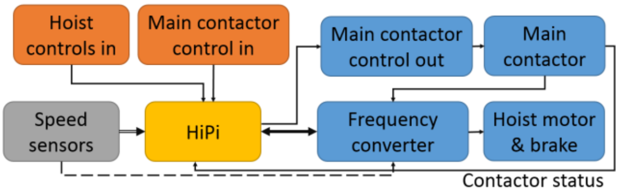

Figure 8. *Architecture for hoist overspeed protection: Hoist with frequency converter control. The bolded line represents safe communication over fieldbus. The speed information can be supplied to the frequency converter for control purposes.*

The basic architecture is very similar to the overload protection architecture for a frequency converter driven hoist. In case an overspeed is detected, the system is brought to a safe state, which means that the hoist movement is halted by stopping the motor and applying the brake. The communication between the HiPi and the frequency converter can be implemented using safe communication over a fieldbus or using digital IO.

The most typical situation where overspeed occurs in hoist drives is when the load control is lost and it is accelerated by gravity. The reaction time is thus very critical in overspeed protection, since the stress to the hoist brake increases to the second power as the load accelerates.

The speed sensors used for hoists are mostly rotational encoders. The typical outputs for encoders are either analog complementary sine and cosine signals, digital pulsing signals or fieldbus interfaces. For an architecture with dual encoders, either two differential analog or digital inputs are needed. For a single safety encoder solution, the communication between the HiPi and the encoder needs to be implemented using a protocol like CANopen Safety.

The required IO for this application is very similar to the overload protection. The speed sensors either require analog inputs for receiving the sine and cosine waves, which will require four analog inputs in total for receiving the two sinusoidal signals from each encoder. If encoders with digital pulse interfaces are used, then 4 digital inputs are required. The third option will be to use a fieldbus to interface the encoders.

An example list of required interfaces for the core functionality of this application architecture would be:

- Dual speed sensor: 4 analog or digital inputs (2 sinusoidal/pulse signals per encoder), or
- Single safe speed sensor: Fieldbus interface for safe communication
- Hoist & main contactor controls: 4 digital inputs (similar to overload protection)
- Feedback from main & brake contactors: 2 inputs
- Communication to frequency converter: Fieldbus interface

3.1.3 Multiple trolley functionality

Often cranes are built such that multiple trolleys are operating on a single crane bridge. In these applications, multiple HiPis could be utilized for implementing more advanced safety functions. Each trolley could include one or more HiPis for supervising the Hoist and Trolley drives of each trolley. The devices can communicate to each other using a safe protocol over a fieldbus, as illustrated in Figure 9, and similarly as already discussed in chapter 2.3.

Safety functions, which could be implemented using such an architecture include:

- Bridge overload: Combination of two or more HiPis with load sensors used to detect when the maximum allowed loading on the bridge in total is exceeded.
- Decentralized collision detection: Combination of two or more trolley drives with safe absolute position detection using individual HiPis, which supervise the position of each other to prevent collisions.
- Tandem running of multiple trolleys: Similar as above, but the devices supervising individually that the relative distances between each other remain within set tolerances.

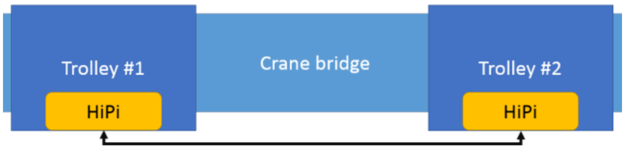

Figure 9. Trolley-to-trolley communication

For a low-cost device, high-speed tandem operation and collision detection will likely not be possible to implement. It is a point for further research to see what would be the limits of the individual devices and their intercommunication for such applications.

3.2 Other safety functions

Other possible safety functions to be implemented in different applications in which the HiPi can be utilized are described in the following chapters.

3.2.1 Electrical power drive systems

Similar to the crane functionalities, but also usable in generic applications, are the electric drive motion safety functions according to [DIN07]. These monitor and/or control adjustable speed electrical power drive systems reacting to given limit values or failures of the system. These values can refer to position, speed, acceleration or torque. As an example of such a safety function, Safely-limited speed (SLS) will be introduced: this function prevents the motor from exceeding the specified speed limit. This can be implemented in, for example, danger areas of a machine such as conveyors, paper machines or winders, in which the operator feeds the material manually. So, the machine does not need to be stopped, but the speed will be decreased and the hazard can be reduced or eliminated.

In Figure 10, the components required for the safety function are shown. A position switch can be implemented as a sensor for detecting when the door to the machine is open, what activates the reduction of the machine's speed. Another types of sensors that could be implemented are light curtains, laser scanners, photoelectric sensors, camera systems, etc. The sensor's signal that activates the safety function, in this case reducing the speed, goes into the HiPi; the monitoring and controlling of the speed takes place here. The HiPi supplies a speed reference to the frequency converter, which supplies the motor, which work together as actuators. An encoder is used to measure the motor rotational speed, and used as a feedback for the HiPi for the implementation of the safety function and for the frequency converter as a feedback for controlling the motor. The HiPi can thus adjust the speed reference when required and supervise that the actual speed is within safe limits.

Other typical motion safety functions are the stopping functions, e.g. Safe torque off, Safe stop 1 and Safe stop 2. These assure that the motor is stopped safely, depending on the machine or application. These safety functions are analogous to the stop categories defined by [DIN14c], which

DOI: 10.2195/lj_Proc_holzweissig_en_201610_01
URN: urn:nbn:de:0009-14-44720

describe the mode in which a machine is to be taken into standstill.

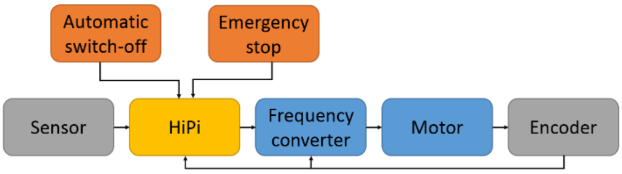

Figure 10. Architecture for power drive systems application

3.2.2 Forklift truck

A significant cause of fatal accidents with forklift trucks is tipping over [Baua12]. Most of the times, this is due to the bad stability of the forklift truck, which is mostly dependent on its mechanical construction.

In order to protect a forklift truck from tipping over, the weight of the load and its position and speed during the movement can be monitored and eventually regulated. These parameters can be monitored with the safety functions for electric drives explained in chapter 3.2.1. For example, the weight can be measured with the torque of the elevating motor for electrically driven system, as the current is directly proportional to the load. For hydraulically driven systems, the weight can be measured from the hydraulic pressure. The position (height) of the load can be determined via a linear encoder. The safety function for the speed of the forklift truck can be realized like presented in chapter 3.2.1. These parameters can be combined with each other, so that the maximum allowed speed depends on the weight and the height of the load.

Stopping the forklift truck as a reaction when the limit value is exceeded is realized under control, which means that the actuators are active and the power to them is removed once the motion has been driven to stop.

4 Conclusions

In this work a concept for a small safety controller, especially designed for intralogistic applications, has been discussed. A hardware, software and communication concept were described, which are designed with a principle of separation between the safety- and non-safety-related parts of the controller. The certification of the safety-related functionality of the controller is done separately, and changes or updates to the non-safety-related parts will not pose requirement for re-certification. The controller is targeted for use in applications, where only small space available, such as the electrical cabinets of small hoists or for AGVs.

5 Future research work

This work only presented the initial concept for a new safety controller. As a future research work the concept will be implemented in practice. The design and creation of the device must be done according to standards [DIN11] and [DIN16a] to be able to obtain a safety certification.

The resulting device should be a flexible research platform. Possible future working topics include researching the achievable reaction times and intercommunication times with such devices, plug & play safety systems and security of internet-connected safety devices.

Literature

[Baua12] Bundesanstalt für Arbeitsschutz und Arbeitsmedizin: *Tödliche Arbeitsunfälle 2001 – 2010*. URL www.baua.de/Toedliche-Arbeitsunfaelle. Referred on 5.5.2015.

[BMWi16] Bundesministerium für Wirtschaft und Energie: *Industrie 4.0: Digitalisierung der Wirtschaft*. URL http://bmwi.de/DE/Themen/Industrie/industrie-4-0.html. Referred on 12.8.2016.

[DIN07] DIN EN 61800-5-2:2007: *Adjustable speed electrical power drive systems – Part 5-2: Safety requirements – Functional; German version*. Beuth.

[DIN10] DIN EN 14492-1:2010-06: *Cranes - Power driven winches and hoists - Part 1: Power driven winches; German version*. Beuth.

[DIN11] DIN EN 61508:2011-02: *Functional safety of electrical/electronic/programmable electronic safety-related systems – Parts 1-3; German version*. Beuth.

[DIN13] DIN EN 13135:2013-05: *Cranes - Safety - Design - Requirements for equipment; German version*. Beuth.

[DIN14a] DIN EN 15011:2011+A1:2014: *Cranes - Bridge and gantry cranes; German version*. Beuth.

[DIN14b] DIN EN 61131-3:2014-06: *Programmable controllers - Part 3: Programming languages (IEC 61131-3:2013); German version*. Beuth.

[DIN14c] DIN EN 60204-1:2007: *Safety of machinery – Electrical equipment of machines – Part 1: General requirements; German version*. Beuth.

DOI: 10.2195/lj_Proc_holzweissig_en_201610_01
URN: urn:nbn:de:0009-14-44720

[DIN16a] DIN EN ISO 13849-1:2016-06: *Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design (ISO 13849-1:2015); German version*. Beuth.

[DIN16b] DIN EN 62061:2016-05: *Safety of machinery - Functional safety of safety-related electrical, electronic and programmable electronic control systems (IEC 62061:2005 + A1:2012 + A2:2015); German version*. Beuth.

[DIN16c] DIN EN 50325-5:2016-06: *Industrial communications subsystem based on ISO 11898 (CAN) for controller-device interfaces - Part 5: Functional safety communication based on EN 50325-4; English version EN 50325-5:2010*. Beuth.

[EU06] European commission: *Directive 2006/42/EC on machinery*. URL http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:157:0024:0086:EN:PDF. Referred on 12.08.2016.

[Hay14] A. Hayek, B. Machmur, M. Schreiber, J. Börcsök, S. Gölz and M. Epp: *HICore1: "Safety on a chip" turnkey solution for industrial control*. 2014 IEEE 25th International Conference on Application-Specific Systems, Architectures and Processors, Zurich, 2014, pp. 74-75.

[NXP12] NXP Semiconductors: *Functional safety for industrial applications*. 2012. URL https://cache.freescale.com/files/industrial/doc/brochure/BRFNCSFTYIND.pdf. Referred on 6.8.2016.

[Pla97] P.J.Plauger: *Embedded C++: An Overview*. Embedded systems programming. 1997.

[Renesas16] Renesas Electronics Corporation: *Functional safety solution for industrial automation*. 2016. URL https://www.renesas.com/en-us/solutions/factory/common-technologies/functional-safety-solution-for-industrial-automation.html. Referred on 6.8.2016.

[TI16] Texas instruments: *SafeTI Design packages for Functional Safety Applications*. 2016. URL http://www.ti.com/ww/en/functional_safety/safeti/index.html. Referred on 6.8.2016.

[Tre13] A. Trenkle, Z. Seibold, T. Stoll and K. Furmans: *FiFi – Steuerung eines FTF durch Gesten- und Personenerkennung*. Logistics Journal, Vol. 2013. URL http://www.logistics-journal.de/proceedings/2013/3768. Referred on 9.8.2016.

[WGTL15] S. Stepanyuk, K. Krivenkov and R. Bruns: *Untersuchungen der Gewichtsreduktionspotentiale eines Gegengewichtsgabelstaplers mithilfe aktiver Systeme*. 11. Fachkolloquium der Wissenschaftlichen Gesellschaft für Technische Logistik (WGTL) 2015, pp. 119-125.

Dipl.-Ing. Peter Holzweissig, is working as a Research associate at the chair of Sichere Mechatronische Systeme der Intralogistik (SIMESI), Institut für Fördertechnik und Logistiksysteme (IFL), Karlsruher Institut für Technologie (KIT).

Email: Peter.Holzweissig@kit.edu

M.Sc. Tommi Kivelä, is working as a Research associate at the chair of Sichere Mechatronische Systeme der Intralogistik (SIMESI), Institut für Fördertechnik und Logistiksysteme (IFL), Karlsruher Institut für Technologie (KIT).

Email: Tommi.Kivelae@kit.edu

Ing. Silvia Vélez León, is working as a Research associate at the chair of Sichere Mechatronische Systeme der Intralogistik (SIMESI), Institut für Fördertechnik und Logistiksysteme (IFL), Karlsruher Institut für Technologie (KIT).

Email: Silvia.Velez.Leon@kit.edu

Prof. Dr.-Ing. Markus Golder, is the head of the chair of Sichere Mechatronische Systeme der Intralogistik (SIMESI), Institut für Fördertechnik und Logistiksysteme (IFL), Karlsruher Institut für Technologie (KIT).

Address: Institut für Fördertechnik und Logistiksysteme (IFL), Gebäude 50.38, Gotthard-Franz-Str. 8, 76131 Karlsruhe, Deutschland. Tel.: +49 721-608-48621, Fax: +49 721-608-48629