

DEVELOPMENT OF A REMOTE-CONTROLLED COUPLER-INTERLOCK FOR THE XFEL ACCELERATOR MODULE TEST FACILITY (AMTF)

S. Kotthoff, A. Goessel, C. Mueller, DESY, 22603 Hamburg, Germany

Abstract

For the series production of XFEL cryo modules, it is planned to test all complete modules prior the installation in the XFEL tunnel. For the test stands in the Accelerator Module Test Facility (AMTF), we have developed a new technical interlock. The interlock allows the observation of 8 individual analog signals per device (like temperature, pressure, light, free electrons) and can be fully remote controlled. Each channel has its own upper and lower remote controlled threshold. The channels will be customized with sensor specific module cards, which adapt the power needs and signal conditioning to the specifications of the different sensors. The local display is used as user interface and the little space requirement makes the device quite suitable. An overview of functionality, internal features, connectivity and extendability of the device is given.

MOTIVATION

In AMTF we need several new technical interlocks. Each of them, depending on the test stand, should have between 2 and 69 channels for different sensor types. In FLASH we are using up to now a modified interlock system from HERA, which works very stable, but there is several additional hardware needed (a VME (Versa Module Eurocard) crate with digital IOs (Input Output), slow and fast ADCs (Analog-to-Digital Converter)), to get it partly remote controlled. It requires still an operator in the field, if we like to adjust an interlock threshold.

Thus we decided to build for AMTF a more flexible system, which can be easily adjusted to the different test stands and needs to be available soon. A first proposal of the interlock configuration for a module test stand in AMTF can be seen in Fig. 1. There will be also vertical test stands (working with CW (continuous wave) RF (radio frequency)) in this hall, but there we need much less interlock channels (usual only the vacuum pressure and a He level). We like to have an interlock, that fits to all of these environments.

REQUIREMENTS

As mentioned there is an existing interlock with some features, which we will not miss in the new one:

- It uses one module card for each channel/sensor for the signal conditioning, which makes it very flexible.
- This card takes care also for the power requirement of the sensor. E.g., a light detector will get $\pm 15V$; a PT100 will get a constant current, and so on.

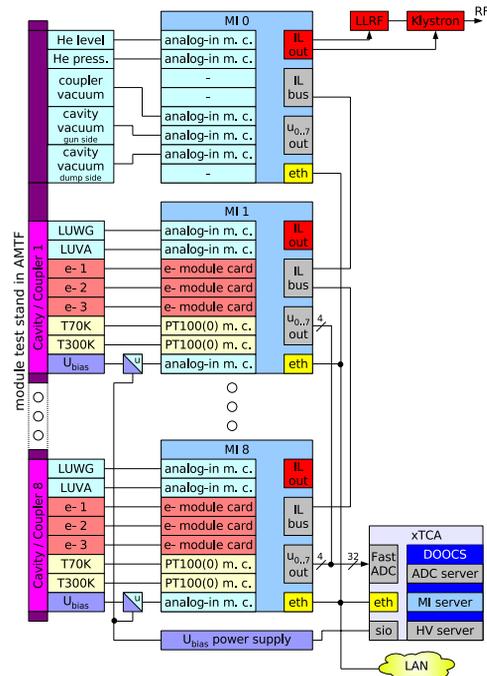


Figure 1: Block diagram of the interlock configuration for a module test stand in AMTF.

- Each input signal will be filtered and amplified in analog technic. The threshold is a defined voltage level, which can be easily used with an analog comparator in order to generate an alarm, if the input exceeds its limits. For this reason there is no dead time between samples, and there is no external clock or triggering needed.

The big disadvantage up to now is the missing remote controll and the many external devices which are needed, in order to get some informations into the digital side of data processing. Here now the needed new features of the interlock:

- It has to be fully remote controlled!
- There should be a test function, which makes it possible to check that the sensor is functional including the signal conditioning. It should no longer be necessary to enter an accelerator tunnel, dismount a sensor and feed in some light (or whatever it likes), only to be sure that it is still working right.
- There should be no software included in the interlock logic. Original we considered to build it in standard TTL (HCMOS), but we realize very early that it will

not be possible to be small in size. Thus we use a complex programmable logic device (CPLD), which works without an external clock signal!

- As much as possible, the parts of the interlock should be checked by the device itself.
- The device must be easily extendable. In the ideal case, the devices can be directly connected together, without the need of different hardware for this combination.
- There must be some diagnostic possible on the device itself, to be fast if something doesn't work.

In the first steps, we expect only some channels are needed, and we are more focused on the adhoc interlocking somewhere in the field. For this application we like to have a standard connector for sensors (6 pins + shield), which can be connected directly. For the more fixed installations like in AMTF we use normally a distribution box. So we planned to build two different backplanes, one with all sensor signals in one connector, and one with the mentioned individual sensor connectors.

DIAGNOSTICS

The interlock device can be divided in two parts: The signal conditioning, comparison and the logic of the interlock, which together takes care about the observed sensors/components. The second part is the microcontroller, which configures all the thresholds, module cards, interlock logic and communicates with a DOOCS [3] (Distributed Object Oriented Control System) server via ethernet. The microcontroller is not involved in the interlock function itself, thus in general it is possible to restart it, without any influence on the active running interlock.

Diagnostics Local on the Device

On the front panel there are LEDs (Light Emitting Diode), which are showing in realtime the state of the interlock logic. For each channel there is an upper and lower alarm LED (red) and also an upper and lower mask LED (yellow), if such a limitation is not used. A seven segment display shows which alarm channel comes up first. There is also a LED which display if this device was the first one in the group of interlocks or if there is a different device which is the reason (via the lock-line) that this one rises the alarm output. Additionally, there are reset buttons, which can reset only this interlock, or with the second button all interlocks in the same RF group (if there are more than one combined together).

The microcontroller is using a four line LC-Display, which allows the operator to read individual channel states (including the measured value and thresholds per channel in there calibrated real units, like a temperature in Kelvin shown in Fig. 2). Also all status bits are shown, in order to

get out, if an IC is not answering on one of the i²c (Inter-Integrated Circuit) buses, or a system voltage is missing. The network configuration and status can be seen, which is the most usefull point, if the device cannot be accessed via the network (for example if there is a bad/wrong DHCP (Dynamic Host Configuration Protocol) server acting in the same network segment). (The device can be also configured to use static network addresses.)

It is not possible to change any settings through the LCD (Liquid Crystal Display) menu, this has to be done through the network connection.

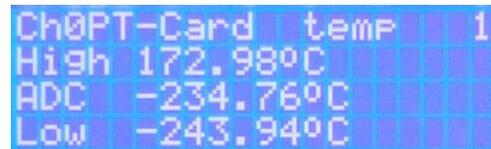


Figure 2: Example of the LC-Display: A measured temperature and its high and low alarm limits.

Diagnostics Remotely via Ethernet

What informations we like to know from the device? That depends; roughly there should be somewhere a DOOCS panel, which displays the actual signal values and their alarm states.

If there is a problem with this simple informations, there are many possibilities, what can go wrong. All states which are shown via the LEDs (interlock state) and are visible in the LC-Display menu are also accessible via ethernet. The "data-set" structure allows to see the status of any IC on the i²c buses, the set and readback values (if available) and also if the setting succeeds.

Additionally, there is the possibility to set values via ethernet, therefore are two different accounts created. The normal user can set all interlock relevant settings, like thresholds, channel masks and manual alarm. The admin account has additionally the permission to change the interlock channel calibration or configuration and all system relevant settings (like network, module card power, processing on the i²c bus, ...). If there is any internal communication channel not working, it will be shown in a sum status byte, which is returned on any communication access.

WORKING PRINCIPLE

In the block diagram (Fig. 3) the main functional groups of the device are shown. The logic of the CPLD will get all individual alarm signals (two per channel) and also the alarm state of the other interlock devices (via the lock-line). It will get the reset information from the front-panel buttons and also from other devices (via the reset-line). The status will be shown directly with several LEDs.

The microcontroller handles the network interface and measures the analog values with an internal 12 bit ADC directly. It can communicate with the CPLD, the DACs

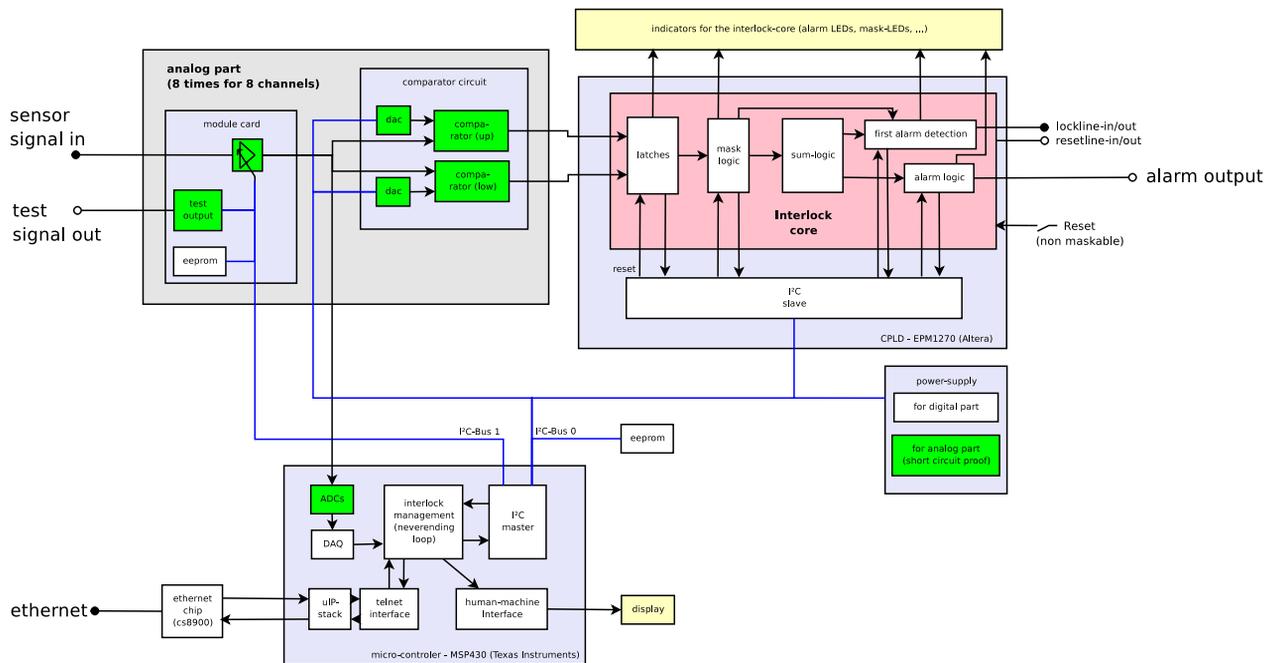


Figure 3: Block diagram of the interlock.

(Digital-to-Analog Converter) (for the alarm thresholds) and the module cards through two independent i^2c buses. It is also able to power off any module card (channel) and also the CPLD (which would generate an alarm). Up to now there was no reason to touch the CPLD power, but the channel power should be manageable, because a sensor can short there power supply. If that is the case we can switch off this channel completely and run the interlock with the other channels only. If there would be enough to operate the system in a safe manner.

Interlock Logic

The logic is simple, each input channel has a latch, and can be masked. It will be latched additionally, which channel alarm rises first. The external lock-line will be pulled down (=active), so that other interlocks will activate there alarm outputs too. The klystron needs to be connected on one interlock of a group, only. Also between the interlock devices, the first one will be remembered and all of them are latching there output signals additionally, so that all of them need to get a reset for enabling the RF again (if the reason of the alarm was gone). To be complete, also the reset is wired between all devices, so that it is enough to send only one of them a group reset.

The CPLD contains an i^2c slave, for the communication with the microcontroller. All channel inputs can be read, the masks can be read and write, the channels can be reset, and a manual alarm can be generated also.

One important point for the VHDL (Very High Speed Integrated Circuit Hardware Description Language) code inside the CPLD is, that we avoid to use a additional clock signal. That means the interlock logic is only state driven.

08 Ancillary systems

Firmware

We used the mspgcc [1] toolchain in order to build the firmware of a MSP430F1612 Texas Instruments microcontroller, which takes over the management of this device. It uses the uIP [2] TCP/IP stack for the ethernet communication with the corresponding DOOCS server.

All communications are interrupt driven, so that the device management is running more or less free in a permanent loop. Thus that it will not be locked by incomplete communications. There are three big configuration structures defined in the firmware.

- One of it takes care about the configured data transport (called data-set), which works like the following: If a new parameter is defined by the external server (like a new module card setting), it will be written to the hardware and stored in the memory of the microcontroller in parallel. Additionally the settings will be read back periodically and compared with the value, which was already set before. If it is wrong, it will be written again and an error bit will be set. Any wrong behavior (communication errors, lost information in a module card, ...) will be reported in individual status flags for each IO channel (like an i^2c slave chip), which is also visible in the main/sum error byte. The main error byte is returned after any external access, so that the more detailed error bits can be requested from outside if it is necessary only.
- The i^2c -set contains the protocol of each i^2c chip and knows also the protocol header to switch the bus to the right module card.

- The menu-set contains the function calls in order to display the human readable informations on the LCD. It knows also the actual functions of the buttons (up, down, ok, back). It distinguishes between menu displays and function displays. The function displays are able to update the shown informations (like ADC values) without a pressed button.

The firmware provides not only commands for the normal interlock operation. There are also many commands for detailed diagnostic as well as the programming of module cards and the system configuration is foreseen.

MODULE CARDS

The designed module card interface is identical for each device channel, so that the cards can be mixed like it is needed. The sensor connector uses always 6 pins, which are defined by the card only, and should contain in general the signal to measure, the power supply for the sensor and a test signal from the module card in order to test the right function of the sensor.

The digital part of the card offers 8 free defined bits, which are used for switching between different low pass filters, gains, power supplies, to enable the test function or to enable a LED on the module card which can be used by the DOOCS server to mark a channel for the maintenance. The memory on the card is used to store the ID, the card label, calibrations and the labels of the 8 individual configuration bits. The bit labels are used to show a readable state of each bit in the display of the device.

Actual we build 3 different module cards.

- PT100(0) module card. The test function reduces the constant current of the sensor.
- Analog-In module card, with a differential input for $\pm 1V$ and $\pm 10V$. It supports the sensor with $\pm 15V$ and a test output with $+15V$. A low pass filter can be enabled.
- The e- module card is similar to the Analog-In type, except that it measures the current on a 100 Ohm shunt and contains a floating voltage source (0V, $\pm 15V$, $\pm 30V$).

The analog main board is able to switch off the power of each channel individually, if a external sensor is shorting the power supply.

OUTLOOK

A series prototype of this interlock is just build and will be tested as soon as possible. All cavity and module test stands for the XFEL will use this interlock type. Only the fast sensor channels need additional external ADCs (1MS/s), which are available from the controll system (DOOCS). The system allows to be as flexible as possible, also during the cavity production it would be easily possible to change the type of a channel.

08 Ancillary systems

REFERENCES

- [1] mspgcc: A port of the GNU tools to the Texas Instruments MSP430 microcontrollers done by Chris Liechti and Dmitry Diky. <http://mspgcc.sourceforge.net/>
- [2] The open-source uIP TCP/IP stack was developed by Adam Dunkels of the Networked Embedded Systems group at the Swedish Institute of Computer Science. http://www.sics.se/~adam/uip/index.php/Main_Page
- [3] K. Rehlich et al, DOOCS: an Object Oriented Control System as the integrating part for the TTF Linac, Proceedings ICALEPCS '97, Beijing, China. <http://doocs.desy.de/>