

The CMS Tracker Control System

Alexander Dierlamm, Guido H. Dirkes, Manuel Fahrner, Martin Frey, Frank Hartmann, Lorenzo Masetti, Otilia Militaru, S.Yousaf Shah, Robert Stringer, Andromachi Tsirou

CERN, 1211, Geneva 23

E-mail: lorenzo.masetti@cern.ch

Abstract. The Tracker Control System (TCS) is a distributed control software to operate about 2000 power supplies for the silicon modules of the CMS Tracker and monitor its environmental sensors. TCS must thus be able to handle about 10^4 power supply parameters, about 10^3 environmental probes from the Programmable Logic Controllers of the Tracker Safety System (TSS), about 10^5 parameters read via DAQ from the DCUs in all front end hybrids and from CCUs in all control groups. TCS is built on top of an industrial SCADA program (PVSS) extended with a framework developed at CERN (JCOP) and used by all LHC experiments. The logical partitioning of the detector is reflected in the hierarchical structure of the TCS, where commands move down to the individual hardware devices, while states are reported up to the root which is interfaced to the broader CMS control system. The system computes and continuously monitors the mean and maximum values of critical parameters and updates the percentage of currently operating hardware. Automatic procedures switch off selected parts of the detector using detailed granularity and avoiding widespread TSS intervention.

1. Introduction

The silicon strip tracker of the CMS detector consists of 15232 modules, built of 24328 silicon sensors covering a surface of 206 m^2 . The main task of the Tracker Control System (TCS) is the safe switch on and off of the detector, taking into account the information coming from the Tracker Safety System and from the Detector Control Units of the individual modules.

During an extensive test of a significant part of the Tracker, an advanced prototype of the control system has been successfully used, providing a necessary feedback for the further development of a stable and usable software, which will be used for the years of operation of the CMS experiment.

2. Hardware

2.1. Power Supply System

The Power Supply System for the CMS Tracker is delivered by CAEN, Viareggio (Italy). Its building block is the Power Supply Unit (PSU) providing two low voltage sources (1.25 V and 2.5 V) and two high voltage supplies to a group of 6 to 15 modules. Two PSUs are combined into one Power Supply Module (PSM) of type A4601H. The power to the control ring is delivered by a separate set of PSMs (A4602, providing 4 low voltage channels). The modules are grouped in 1944 power groups and 356 control groups (each control group contains several power groups that share the same control power channel). Up to 9 PSMs are installed inside one crate, up

acknowledged in the PLC system to release the interlock from the relays. Then the power supply system has to be cleared with a command given at the level of the mainframe and the power can be switched on again.

The Tracker Control System reads the environmental measurements from the PLC for displaying, archiving and analyzing purposes. Automatic safety procedures in the control system can avoid the widespread intervention of Tracker Safety System (TSS) interlocks by switching off critical parts of the detector with a much finer granularity.

2.3. Detector Control Units and Communication and Control Units

Each module of the CMS Silicon Strip Tracker has a custom integrated circuit called DCU (Detector Control Unit) which measures power supply voltages, leakage current and temperatures.

A similar circuit, called CCU (Communication and Control Unit) sits on the control ring and provides measurements with the granularity of the control group.

DCUs and CCUs are read out by the Tracker DAQ system and sent to TCS. The calibration is performed by DAQ while the control system should analyze, group, display and take actions depending on the converted values of the DCU.

3. Software architecture

3.1. Software layers

The control system is implemented using a commercial SCADA¹ software, ETM PVSS², that has been extended at CERN by a common LHC framework (JCOP³[4]) in order to adjust to the needs of a HEP experiment and to gain from overall developments. The specific Tracker libraries are built on top of PVSS and of the framework.

The communication with the hardware uses standard protocols; in the case of the Tracker, OPC⁴ is used for communication with the power supply system and S7 (Siemens) allows the communication with Siemens PLC. PVSS provides the drivers for these two protocols while CAEN provides an OPC server that exports the power supply system parameters.

This way, a layer of abstraction is provided to the control system developer, who can therefore focus on the control tasks rather than on the specific hardware features. PVSS drivers update on change the value of the data in the central event manager and can apply a smoothing, in order to ignore small uninteresting fluctuations in the measurements.

3.2. Hierarchical control

The logical partitioning of the detector is reflected in a Finite State Machine (FSM) hierarchy, organized in a tree structure where every node has a state and accepts a defined set of commands from the users. The leaves of the tree correspond to the hardware devices, while the internal nodes provide an abstraction level for all the subtree below. States and alarms are propagated from the leaves up to the root while commands move in the opposite direction. The hierarchy is encoded in the SMI++⁵ language, an evolution of a tool originally developed for DELPHI at LEP [1]. The hierarchical approach facilitates summary information, commanding to large parts of the detector and error reporting. The high level actions allow to bring the entire detector to physics state (following the correct switching sequence) by giving only one global command.

¹ Supervisory Control and Data Acquisition

² Prozeß-Visualisierungs und Steuerungs System

³ Joint Controls Project

⁴ OLE for Process Control

⁵ State Management Interface

SMI++ supports the inclusion and exclusion of some parts of the hierarchy. When a node in the tree is excluded, its state is ignored by its parent and the commands from the parent are not sent to the child. It is also possible for several users to control in parallel different portions of the detector. For example, a sub-detector expert can control a subtree while the main shifter has still the control of the rest of the hierarchy.

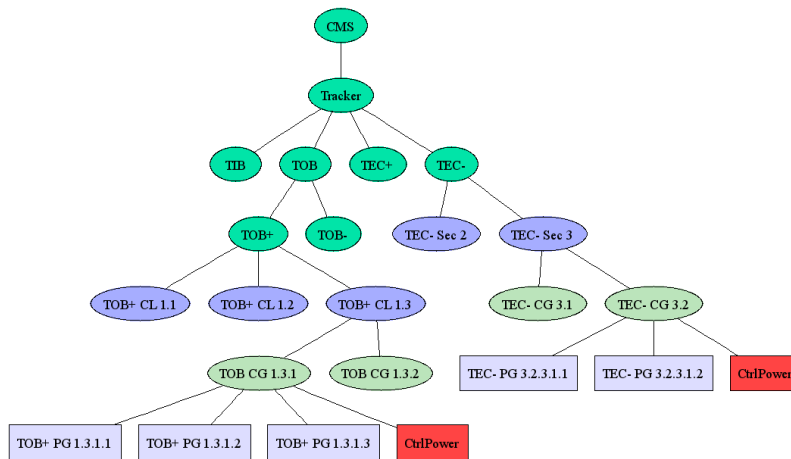


Figure 2. Control Hierarchy for the Tracker. The figure shows a partial hierarchy down to some power groups in the Tracker Outer Barrel and in the Tracker End Cap.

The root of the Tracker control hierarchy is an interface to the broader CMS control system. It is split into four nodes that correspond to the high level trigger partitions in the DAQ system: TIB (Tracker Inner Barrel), TOB (Tracker Outer Barrel), TEC+ (Tracker End Cap on $z+$ side) and TEC- (Tracker End Cap on $z-$ side). The next levels reflect, for TIB and TOB, the division between plus and minus side followed by a layer of cooling loops, while TEC+ and TEC- are partitioned into sectors. Below cooling loops and sectors, the control groups represent the state of a control ring, built by the basic blocks of the power supply system: the control power channel and the power groups (Fig. 2).

The modules of a cooling loop share the cooling lines, so that in case of problems in the cooling system the temperature will increase in the entire loop. The software safety allows to switch off a single cooling loop in response to an anomalous value in the environmental probes related to it. In the case of the TEC, control groups can belong to more than one cooling loop, so the hierarchy must necessarily go directly from sectors to control groups and the software safety acts at the level of sector.

3.3. Task distribution

The CMS Tracker Control System is distributed among several computers, exploiting the distribution capability of PVSS (Fig. 3).

Four machines are devoted to power supply control, each one communicating with one of the mainframes. These machines also manage the bottom layers of the FSM hierarchy. Since the state of the subtree under each cooling loop or sector is computed by a separate process, and each machine is connected to only one mainframe, it is essential that the PSUs belonging to the same cooling loop do not span across two different mainframes. This constraint is satisfied in the cabling scheme of the detector.

One machine works as supervisor, containing the upper layers of the hierarchy and managing the connection to the auxiliary systems, such as the beam condition monitoring.

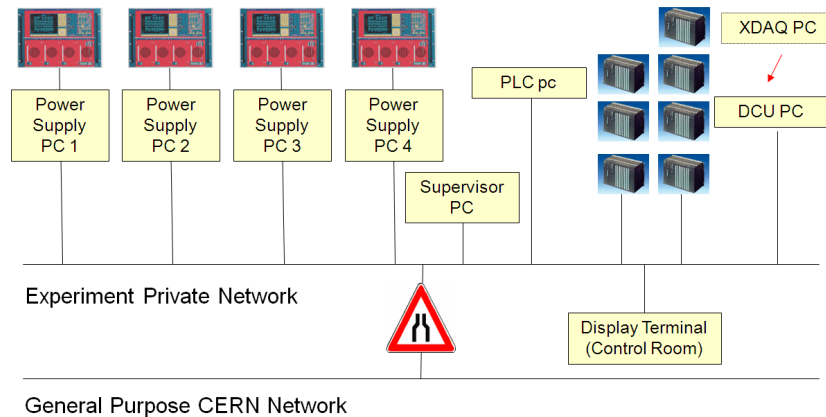


Figure 3. Network for the CMS Tracker Control System

One machine is devoted to PLC reading from 7 independent PLC racks. The control system takes care of the conversion from ADC counts into physical units (i.e. degrees Celsius for temperatures and percentages for humidities), using probe-specific fitting constants which are read on configuration from the Tracker configuration database (see 3.7). Temperatures measured on silicon (for TOB and TIB) or on cooling lines (for TEC) are grouped by cooling loops. For each cooling loop, mean, maximum and minimum values are computed in real time. Dew point values are also computed (from pairs of related air temperature and humidity sensors) and compared to the coldest point in their region of the Tracker. An alarm is raised when a dew point gets too close to the coldest temperature in its region.

The DCU values are managed in a separate machine. In this case the values are not directly read out by PVSS via a driver, but are received from the Tracker DAQ system. A DAQ process polls the DCU values periodically for each High Level Trigger Partition and notifies the TCS using PSX⁶, a SOAP service that allows communication with PVSS using this standard protocol. After the DAQ has processed the DCU data, it sends a command to the control system to update the mean and maximum values of the DCU/CCU elements related to each power group or control group.

All the computers for the control system are in the experiment private network. Communication with the rest of Internet is only possible through dedicated servers.

3.4. On-line updating of percentage of on devices and of mean and maximum values of critical parameters

SMI++ is capable of propagating state changes bottom-up, using logical conditions that define the state of one node depending on the state of its children. However, this approach turned out not to be suited to the Tracker needs. Due to the large amount of hardware, it is not possible for 100% of the Tracker to be always on. Since the DAQ has to wait for the Tracker to be in ON state to take data and stops when the detector exits from this state, it is important to introduce some tolerance and keep on running even if some modules are not powered. For this purpose, the FSM moves from “mixed” to “pure” states when the percentage goes above a certain threshold (typically 95%). For the same reason an error in a power supply channel (e.g. a trip) should not bring the entire Tracker in ERROR state, so the transition to the ERROR state is ruled by another threshold (typically 5%). The states for the nodes above the control group are listed in Table 1.

⁶ PVSS SOAP eXchange

Table 1. Meaning of the states. All the (non empty) conditions in one row must be valid in that state.

State	# Ctrl On	# LV On	# HV On	# Error
OFF	0	0	0	$x \leq 5\%$
CTRLMIXED	$0 < x \leq 95\%$	0	0	$x \leq 5\%$
ON_CTRL	$x > 95\%$	0	0	$x \leq 5\%$
LVMIXED		$0 < x \leq 95\%$	0	$x \leq 5\%$
ON_LV		$x > 95\%$	0	$x \leq 5\%$
HVMIXED			$0 < x \leq 95\%$	$x \leq 5\%$
ON			$x > 95\%$	$x \leq 5\%$
ERROR				$x > 5\%$

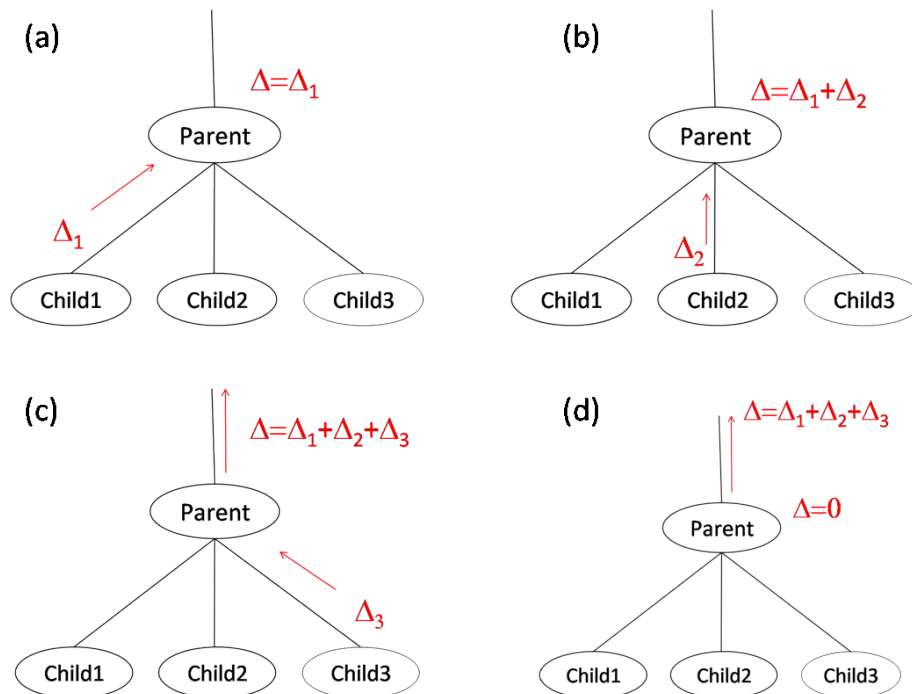


Figure 4. Propagation example for one node with three children: three threads report their changes to the parent (a,b,c). All three threads wait for a timeout, but only the third one finds that the value of Δ in the parent has not changed, so only this last thread will propagate up the change and reset the value of Δ in the parent (d), resulting in a great reduction of the number of concurrent threads.

The real-time computation of the percentage of ON devices is implemented in dedicated processes, running on the computers managing power supplies and on the supervisor. To improve performance, the difference from the previous state is propagated in the tree on change of the state of one device, updating the nodes from the leaf up to the root, without need of reading the values of children that have not changed. This method could however lead to performance problems: when a significant portion of the detector is switched on or off, a large number of state changes of the leaves would result in many updates for the top nodes. Since each update runs in a separate thread, the system could be overloaded because the node updates must be

executed in mutual exclusion. This problem is solved by a simple algorithm that collects the changes coming from the children. Each thread writes the computed difference into its parent node, then waits for a certain timeout. Only if no other thread has changed this value during the timeout, the change is propagated up in the tree, otherwise the thread stops (is “kicked out”). A maximum number of kicked out threads is introduced to ensure that the highest levels are eventually updated in case of continuous fluctuations. This way the number of concurrent threads can be greatly reduced by choosing an appropriate value for the timeout (see Fig. 4). The same algorithm handles the inclusion and exclusion of subtrees from the FSM hierarchy, by updating the total number of devices rather than the number of on devices.

This approach can be easily extended to update on line mean, minimum and maximum temperature for each subtree.

3.5. Safety warnings and automatic actions

Automatic actions are programmed in the control system, based on the environmental data read from the safety system. These actions always go in the direction of avoiding the intervention of the safety system but should never be considered reliable; detector safety is ensured independently by TSS.

Probes are grouped by cooling loop and a dedicated process updates, on change of any temperature, the mean, maximum and minimum value for each group.

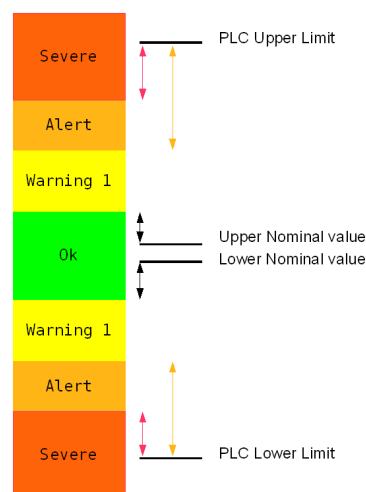


Figure 5. Alarm ranges for the cooling loop: a set of differences (represented by the arrows) allow the definition of the alarm ranges depending on the nominal temperature and the TSS limits

Depending on the nominal interval for the temperature of the specific cooling loop, the hardware limits of the specific probes and a configurable set of ΔT 's (see Fig. 5), warning and alert limits are defined for each cooling loop. The alarms are displayed to the shifter who must be instructed to properly react and to call the expert in case of severe problems.

In each cooling loop three safety limits are defined for minimum, maximum and mean value: minimum and maximum limits correspond to the most severe alarm level (at the configured difference from the corresponding interlock limit), while the limit for mean value is set at twice the distance from hardware upper limit.

When a specific number of sensors (configurable as well for each individual cooling loop) goes out of the limits for maximum or minimum or when the mean value of the temperature goes above the mean limit, TCS issues an emergency switch-off command to the concerned part of

the detector. An audio alarm is played when an emergency switching off action is performed. The software safety also blocks the switching on of the concerned part of the detector until the values go back into the normal range.

A similar safety procedure is implemented based on the values read from the DCUs which have a better granularity, though the precision of the measurement is not as accurate as in the PLC probes. The safety switching off due to the DCUs acts at the level of power groups and control groups and should in principle prevent the intervention of the next level safety. The information from DCUs is only available when the modules are powered, so after a safety switch off it is not possible to get the updated values from the DCUs. Safety actions at the level of DCUs can be enabled and disabled by the user, while the safety based on PLC readings cannot be disabled.

3.6. Historical archiving

Historical values of power supply status, voltages, currents, PLC and DCU readings are written on change to an Oracle database. A dead-band is applied to the archiving in order to archive only significant changes. PVSS is able to retrieve data from Oracle for simple plot displaying, but since data is stored in a database it is possible to use standard tools, such as ROOT, for off-line analysis of the archived data.

3.7. Tracker Configuration Database

The building of the Tracker hierarchy requires the mapping from the hardware (power supplies and environmental probes) to the detector parts. This information is also needed for configuring the PLC logic. Moreover, the correctness of the map is crucial for operation and safety and must thus be accurately checked.

For this purpose, data coming from various and heterogeneous sources has been assembled into a custom designed Oracle database in which all the nodes of the control hierarchy are represented, together with the cabling and configuration information of the hardware. Due to the size and the extreme complexity of the Tracker hardware, it would not have been possible to build the required components without the help of a database.

For PLC configuration, information about sensors, relays and groups has to be combined with the detector parts where the probe is connected and the crate that is interlocked by the relay. In the relational database this information is split into different tables in order to remove the tight coupling of data. Sensor masks and group masks that define the behaviour of the interlock logic are computed from these tables and configured in the PLC hardware by TSS.

After the complete configuration of the system, an image of the system is saved to a general configuration database provided by the JCOP framework that stores all the addressing, archiving and configuration parameters and allows to restore the complete configuration of each computer after any crash.

3.8. Checking procedures

After the configuration of the control system, several checking procedures have to be performed in order to ensure the correctness of the cabling:

- all power supply cables are attached, before connecting the detector, to some test load boxes which allow to verify that the cable is connected to the correct power supply unit and that the environmental values can be read at the expected address in the PLC system;
- after the successful result of the first procedure, the open end of the power supply cables can be connected to the detector. This step is checked by switching on power groups one by one and verifying that the unique identifiers of the DCUs read by the DAQ system match the ones listed in the Tracker Configuration Database;

- the configuration of the interlock system is checked by changing the limits of each sensor and checking that the correct power supply crate is actually interlocked.

Notice that, though the Tracker Safety System is completely independent of the control system during normal running, ensuring its configuration and correctness is a crucial task which must be performed by the control system during commissioning.

4. Graphical User Interface

The graphical interface for controlling such a complex system must achieve the goal of providing a simple interface to non expert shifters but should also allow the expert to act at a very detailed level in case of problems. The user can browse the Tracker hierarchy, view the state and the percentages of on devices below each node and give commands at any level in the hierarchy. In case of problems, the alerts and the global view of the states provide a fast access to the affected part of the detector.

Access control is implemented into the GUI: the users have different privileges depending on their level expertise.

5. Periodical health check of the Tracker

Some critical conditions that must be reported as alerts, such as a tripped channel or a too high temperature, can be detected just in the function of the current state of the hardware. However, there are some critical or anomalous conditions that can only be identified by analyzing the historical trend of the parameter or by comparing it to other related readings.

The monitoring of the interdependent values is implemented in dedicated control scripts that query periodically (typically every 5 minutes) power supply values and environmental probes.

Examples of anomalous conditions that can be detected by the monitoring systems are:

- Switched on power supply channels with zero current;
- Sudden jumps in the temperature or humidity values that are most likely due to malfunctioning probes;
- Rising temperatures that can be detected before they hit a warning level;
- Replacement of Power Supply Modules by looking at the serial number.

This kind of analysis results in alerts, that can be monitored by detector experts, and in SMS or email warnings to the interested persons.

6. Conclusions

A prototype of the software has been successfully used and improved during an extensive test with 25% of the final hardware connected. The experience gained in these months was useful to understand the needs of the users, to improve the performance of the system and emphasized as well the need of a coherent custom database for building the system.

The present prototype of the control system is quite advanced and flexible and will be used, with a few modifications, for all the years of operation of the CMS experiment.

References

- [1] Franek B J and Gaspar C 1998 SMI++ object oriented framework for designing and implementing distributed control systems - *IEEE Trans. Nucl. Sci.* **45** 1946-1950
- [2] Hartmann F for the CMS Tracker Control System Group 2007 The CMS Tracker Control & Safety System, *online contribution to Vienna Conference on Instrumentation VCI 2007*
- [3] Frey M for the collaboration 2005 The CMS-tracker detector control system *Como 2005, Astroparticle, particle and space physics, detectors and medical physics applications* 1067-1071.
- [4] *JCOP Architecture Working Group: Framework Design Proposal*, CERN, Geneva, 2001, CERN-JCOP-2000-008