SMARTFACTORY, A MODULAR, LOW COST PRODUCTIVITY MONITORING SYSTEM

Peter A. Bosscha[1], Piet Terblanche[2]
[1] CSIR, Materials Science and Manufacturing
Pretoria, South Africa
e-mail1: pbosscha@csir.co.za
[2] CSIR, Materials Science and Manufacturing
Pretoria, South Africa
e-mail2: pterb@csir.co.za

ABSTRACT

As a technology colony, South Africa has often had to rely on imported technologies to assist in the monitoring of production floor statistics. This results in high costs in both procurement and support of such systems. Whilst this is the accepted norm for larger corporations, it is simply not feasible for the small, medium and micro enterprises (SMME) where usually a manual monitoring system is implemented. These manual systems have a number of disadvantages such as increased labour costs, a lack of objectivity and plain incorrect data entries. In aid of the SMMEs the Smartfactory system offers a low-cost modular monitoring solution capable of implementation in a step by step fashion with little training required in installation, maintenance and upgrades. The Smartfactory system can be used in many diverse manufacturing environments. This paper presents the background, development and implementation of the system and the experiences had with it so-far.

Keywords: Production monitoring, low cost, SMME aids.

1 INTRODUCTION

The CSIR has a long history of working with South African industry to achieve optimised production processes. Often these efforts take the form of studies, implementation of recommendations in studies and specialised interventions, but in some cases more generic systems are developed. The CSIR was also involved in studies that were performed for the Department of Trade and Industry (DTI) and for the Innovation Fund, the compilation of the National R&D strategy and the creation of the Advanced Manufacturing Technology Strategy (AMTS) [1]. All of the studies and strategies agree on the fact that the production technology in South Africa is lacking in Information and Communication Technology (ICT) systems.

The Mechatronics and Micro-Manufacturing as well as the Sensor Science and Technology groups in the CSIR have over the past years developed some specialised ultrasonic measurement and control instrumentation which was sold initially in South Africa only but is currently marketed all over the world. Interaction has been had with a number of industries whilst trialling these systems, and from this experience the need was noticed in South Africa for a low cost monitoring system for the small and medium enterprises.

With some notable exceptions such as Adroit [2], in general the production monitoring tools available in South Africa consist of imported technologies. This means that

purchase, support, and repairs costs are generally so high as to be prohibitive to many of the smaller industries. There are also a number of factories where elements of the production line are state of the art imported machinery, but no integrated system exists linking these advanced devices to the rest of the production line and/or a central office.
From past experiences and the inputs from the previously mentioned studies and strategies it was decided to proceed with the implementation of a complete low cost monitoring and reporting system that could be used by small to medium enterprises. To achieve this the following targets were set:

a. Complete solution.
   Offer a complete integrated solution, not just hardware or software only solution, which makes it easier to implement and keep control over.
b. Low Cost / Affordable solution.
   This addresses the issue where most of the available technologies are relatively expensive, either in purchase or in operation and maintenance of the system.
c. Modular and scalable system.
   This takes into account that a company might either be very small, or might want to start on a small trial basis to see how the solution might work for them and then grow the system later on.
d. Easy to Implement and Use.
   To avoid the use of highly trained personnel for installation, use and maintenance.

There are several ways of implementing a monitoring system, varying from an expensive solution using standard Programmable Logic Controllers (PLC) [3], fast networking and a supervisory control and data acquisition (SCADA) system [4], to the completely non-technical and unreliable solution of a clipboard with a human operator. In between there are a number of other solutions, which either address the software side of the problem, or a hardware measurement solution.

All the available technologies have their advantages and disadvantages, the intended solution attempts to do away with ambiguous measurements (human operator), whilst keeping things as simple and low cost as possible, still measuring and reporting the relevant data which would help a production environment.

2 HARDWARE APPROACH

At the start of the design of the system, doubts still remained whether or not a standard hardware solution should be purchased and made part of the solution. In the end, factors such as costs of available systems and the lack of control over features and indeed availability of these systems helped decide to design a custom solution. Some general goals for hardware design were set:

a. Create non-complex modules
   Implement basic functionality and if extra capability is required then simply add extra modules.
b. Simple Networking

The use of modules implies a communications strategy, and the requirement here was to make that networking as simple as possible, avoiding complex and “sexy” solutions.

c. Adhere to industry standards
   This meant the use of standard power sources, standard interfaces, not inventing new standards.

On the networking side a slightly modified version of the MODBUS protocol [9] was chosen as this is an open, free and well known standard in industry. All modules that are used in the system connect via an isolated RS-485 [5] interface. This is a slightly more expensive solution than a standard non-isolated RS-485 solution, but has an advantage that damage to the network due to electrical faults and lightning strikes does not propagate further through the system. Up to 32 Modules can be connected on the network which is more than sufficient for the needs of a small to medium enterprise.

The first module designed was named SMARTi, seen in Figure 1, and its job was to interface to existing machinery and sensors without interfering with the operation. To that end the module was designed to have isolated interfaces on all inputs. The end specification is six isolated Analog Inputs with 16 Bit resolution, capable of handling current or voltage input, and twelve isolated digital inputs. All data from these inputs is collected, processed, and transferred via the RS-485 interface. At this stage decisions were made on enclosures for the units; this needed to fit in with the industrial theme but also had to be low cost. Eventually a standard enclosure was selected which fits on a DIN-rail [8], making the unit easily installed and removable.

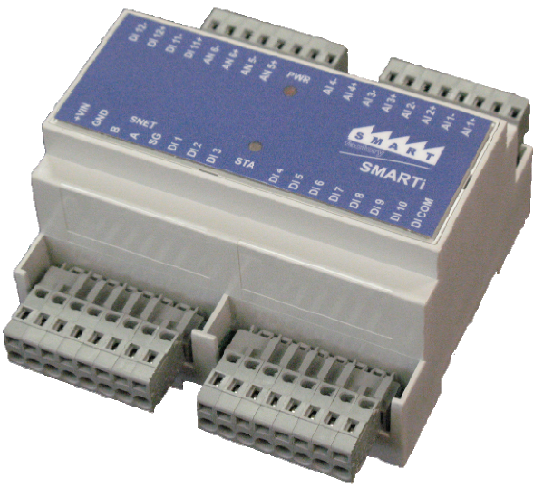

Figure 1: SMARTi unit

The SMARTi unit can be made to interface to almost any type of sensor, as long the signal is a voltage or current output. The unit can be introduced into an existing current loop without disturbing the functioning of the existing machine. During the design phase some arguments were made for the implementation of special interfaces, e.g. thermocouples, but in the end it was decided to leave these specialized functions to standard off-the-shelf transmitters. To make matters easier for installers, use is made of pluggable connectors where the wires are held with a spring loaded clamp mechanism.

This makes it easy to connect wires, but also to remove a unit should this be necessary; simply unplug the connectors without removing the wires and the unit can be replaced.

To interface the network to a PC, again a decision had to be made between designing and purchasing a ready-made unit. In the end the decision fell towards a home-grown solution as it was felt that the available solutions were too expensive and fairly scarce. The second module for this project therefore became a USB to isolated RS-485 interface. The USB interface solution was chosen as this is commonly available on all modern PCs.

The third module was named COMMi, shown in Figure 2, and was created to interface with existing automation. Where SMARTi interfaces to voltages and currents, the COMMi interfaces to entire machines by communicating to their existing PLC's. The COMMi device interfaces to the PLC via a serial interface and translates the information onto the Smartfactory network. The unit also allows for the modification of the values in the PLC's registers. This means that the communication is not only one way; the system is also able to affect the operation of a machine remotely by adjusting operational set-points.

Figure 2: COMMi unit

Having enabled the collection of data in terms of automatically measureable data, the system design was continued in the creation of an interface for a human operator. The HMI unit is intended to capture the events that are not measurable by means of sensors. This unit has an LCD display and a very simple, low button count user interface which is remotely script programmable. The unit allows the capture of data, such as reasons for downtime, which operator is manning which machine and other types of useful data. For reasons of simplicity and familiarity the interface was modeled on the same principles that are used in a mobile phone; menus group the various main options, which then select sub-menus and so forth until the operator makes a selection from an event option. This event is then recorded into the main database the next time the HMI is communicated with.

The HMI also has an additional function as data capture interface; an additional serial port is provided which can be used to interface to other devices such as barcode scanners, RFID scanners or even weighing scales. This then allows for numerous production verification methods, examples are the counting of products on a production line or the registration of a tool in an injection molding machine to record the number of operational cycles.

Figure 3: HMI unit

3 SOFTWARE APPROACH

As one of the targets was low cost, the initial thinking leaned towards the use of a standard Linux distribution such as Ubuntu [6]. The use of the Microsoft Windows Operating System brings with it issues of licensing costs.
However, in communications with industry it became clear that the Windows platform is prevalent, and the use of Linux would result in an extra learning curve and stumbling block for the end user.

The architecture of the system that was implemented in the end is shown in Figure 4. A central database is used for all operations from configuration of the system to data storage to monitoring and reporting, and all software modules of the Smartfactory system interface with this database to perform their functions. A database design was implemented making use of the well known Open Source MySQL database [7]. This solution was chosen as it is a proven solution for many industries, and has a very active community which inspires the confidence that this solution will remain available for a long time in the future.

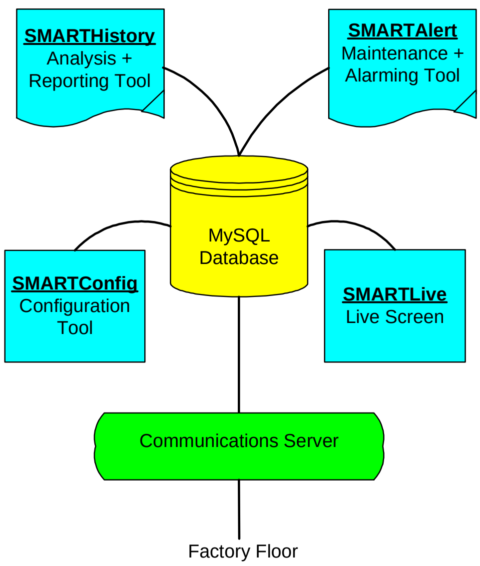

Figure 4: Software Diagram

As a security measure, several access levels are implemented, allowing progressively more access to data and the system. Each software module can be loaded individually on a Personal Computer (PC) that is connected to the company's network, and can be used as long as the Smartfactory database is visible to that software module. This means that for instance "SmartLive" can be used on several PC's whilst on each of these machines completely different aspects of the operation can be monitored.

The communications server is a module that is installed in the background and which communicates with the factory floor via the USB/RS-485 interface module. Its main job is to ensure that the data from all the modules is recorded into the database. Again in keeping things simple and reliable, discovery of units is not performed automatically but rather via the configuration in the database, communication with a unit commences as soon as that unit is listed in the database. The communications server also provides a number of extra services, amongst others it performs updates of calibration files for SMARTi units, script updates for the HMI and automated firmware updates of all units, should this be required. The program is installed as a windows service program and is in general not visible to the user, although an icon is available on the windows toolbar to allow the user to adjust some of the parameters of this process. The use of a service was chosen to ensure that this process is always running, regardless of whether a user is logged in on the machine or not.

3.1 SmartConfig

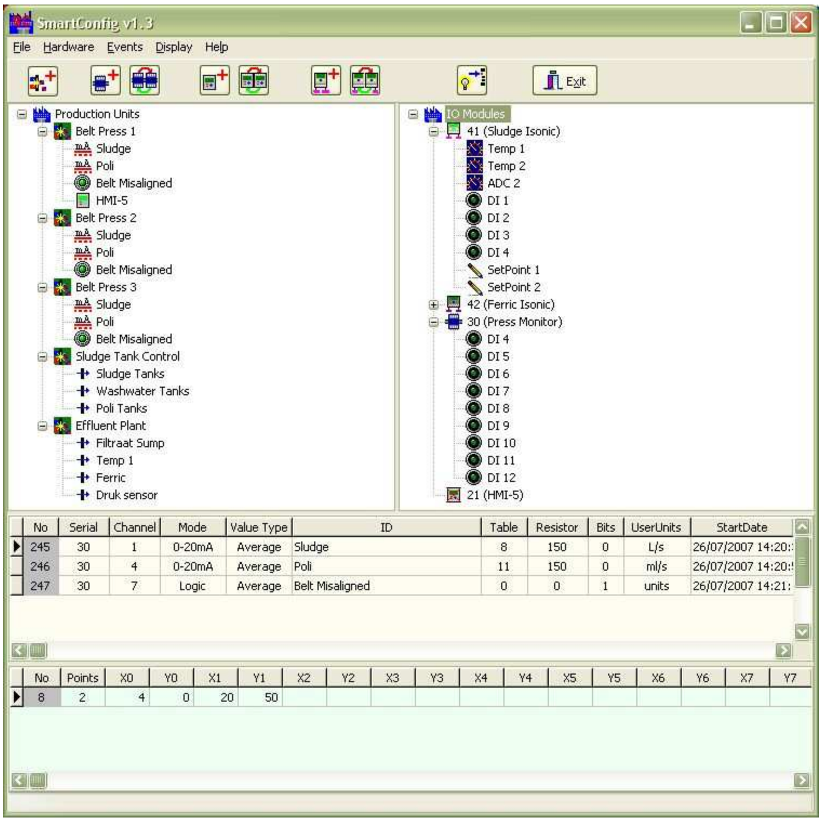

Figure 5: SmartConfig screen

SmartConfig is used to configure the system during installation, updates and changes. The "Production units" tree in Figure 5 represents "machines" or combinations of machines acting as virtual units that do not have to be discreet units physically, but rather logically in terms of production. A production unit could represent more than one physical unit or part of one physical unit or it may be a process. I/O channels are grouped by "Production unit" for display and analysis purposes to not only limit the visual complexity, but also to limit the amount of database traffic.

The “I/O Modules” tree lists physical SmartFactory hardware units, i.e. SMARTis (Analogue and Digital inputs), HMIs (configurable menu for operator inputs) and COMMis (MODBUS communication to third party devices). Each of these units has a unique address and other parameters to be specified as they are added. Note that adding modules to this tree only make them available, it does not "connect" them to the machines.

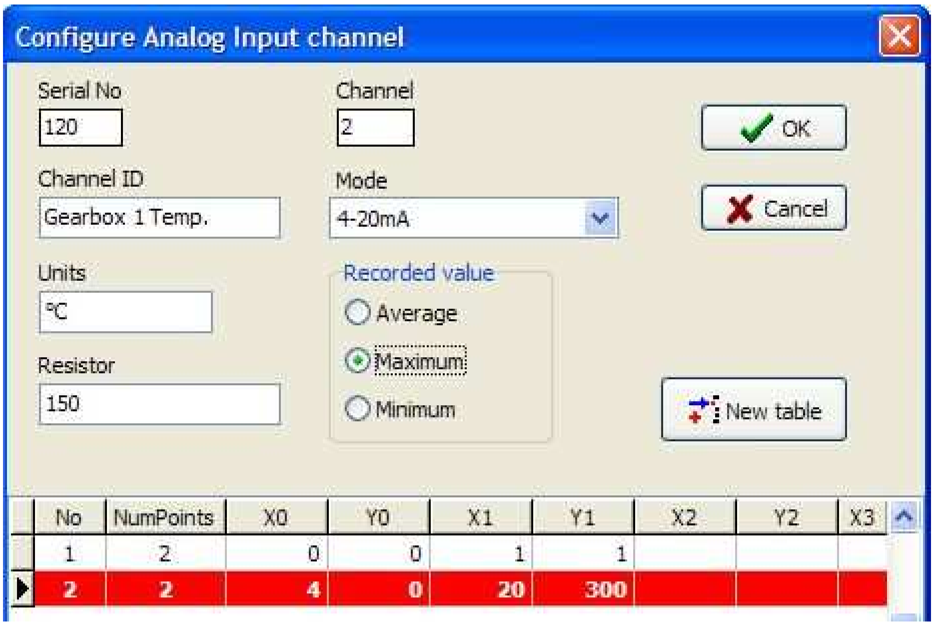

Figure 6: SmartConfig typical channel configuration

To activate a hardware channel, it is dragged from an I/O module and dropped on the target production unit, after which the channel name, type, scaling, etc. is set up. An example of a typical analog channel set-up is given in Figure 6. An interesting variation on digital channel configuration is the ability to combine any number of available digital inputs into a single virtual channel as an n-bit code.

3.2 SmartLive

SmartLive is intended for live representation of only the most recent data, the period of which can be up to a maximum of one week ago. Because this is the SmartFactory module intended for users with the widest range of backgrounds and levels of ICT proficiency, it is fairly rigid and, consequently, quite easy to use without extensive knowledge of the underlying channel configuration.

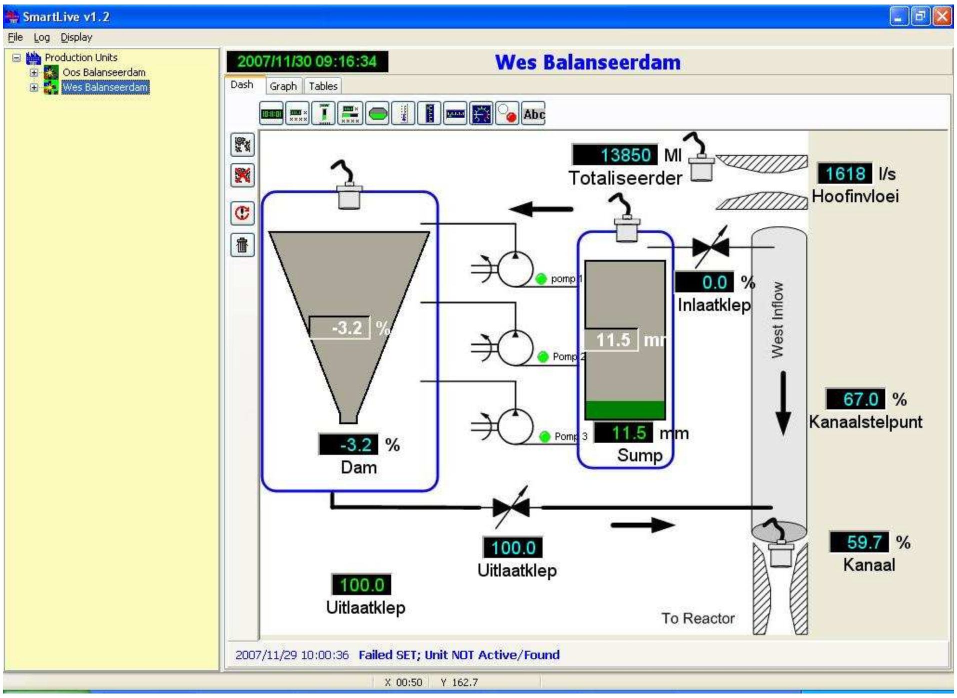

Figure 7: SmartLive, example live Dashboard

Real-time data is presented as a dashboard of indicators, a set of trend graphs and in tabular form. A virtual production unit's easily customizable dashboard is typically updated every 5 seconds, while the trend graphs and tables are updated at the sampling rate (typically one minute interval).

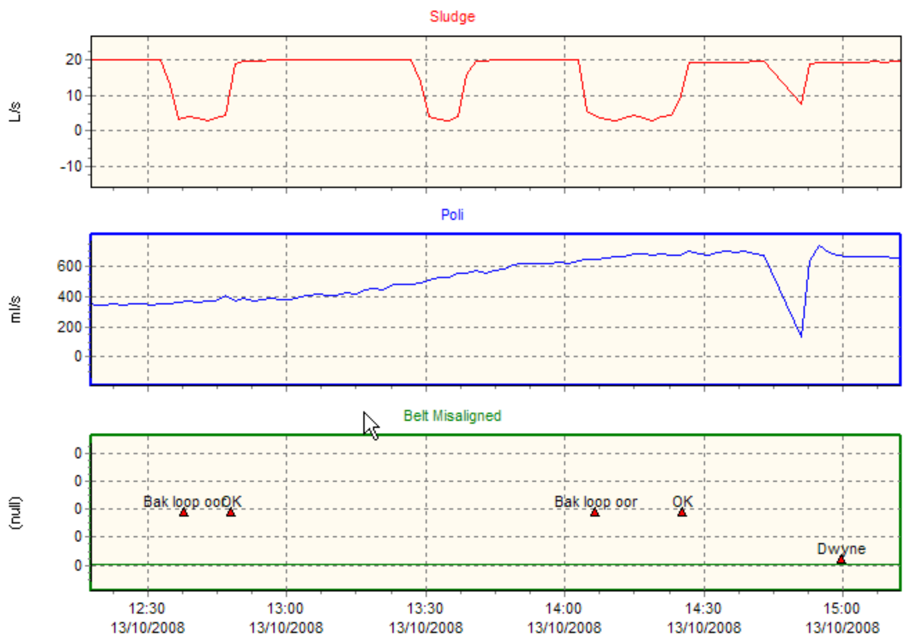

Figure 8: Trend-display in SmartLive

Users can also manually insert "events" via this module. Events are non-periodic time-stamped notes on some or other aspect of the process being monitored, and are more often added to the database via an HMI menu selection.

3.3 SmartHistory

SmartHistory is intended for inspection and post-processing of SmartFactory data. One can inspect configuration and data over any time period and perform limited analysis and/or editing of the recorded data.

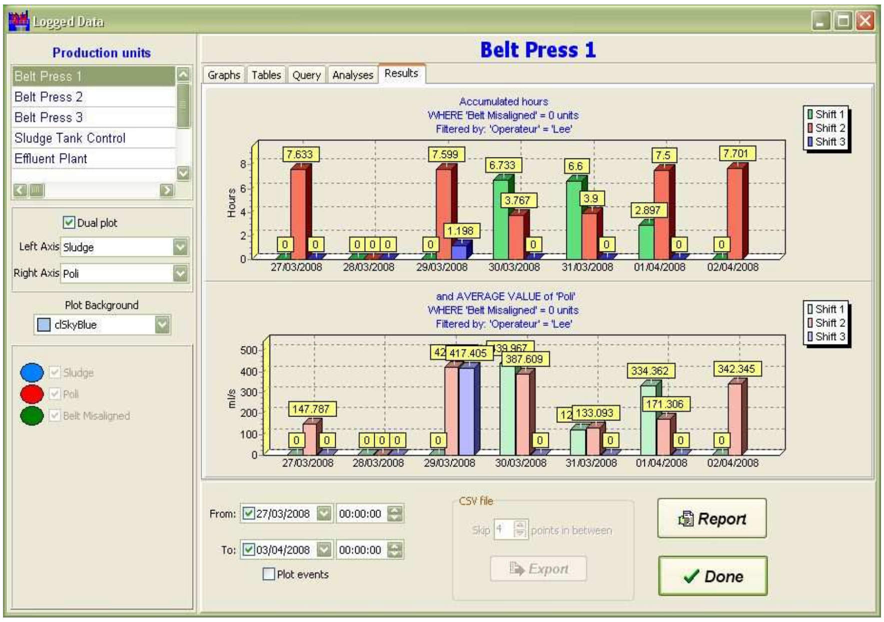

Figure 9: SmartHistory generated report

Common queries are presented in an easily adaptable form for use by non-database specialists. A typical query would be to extract the number of times a certain breakdown occurred during a specific operator’s shift, or to calculate the production per day or shift. More advanced users can compose their own queries using either SmartHistory or the MySQL utility of their choice.

SmartHistory has a number of backup and restore functions for database maintenance, as well as the ability to generate coarser grained long term summaries of selected channels for reports, etc.

3.4 SmartAlert

SmartAlert is used to give visual and audible warning and a database event entry that a measured channel is outside its preset boundaries. Up to eight channels from any combination of machines can be watched at a time.

The SmartAlert philosophy is to allow different users to create individualised alarm levels to suit their own purposes. This way the production manager can be alerted when the average rate at which a certain product is being made falls below a set level, while the maintenance manager can be alerted when lots of short stoppages for fixing equipment cause that same rate to vary widely even though the average is still within bounds.

All calculations are done in a sliding window of specified width which is specified individually for the different channels.

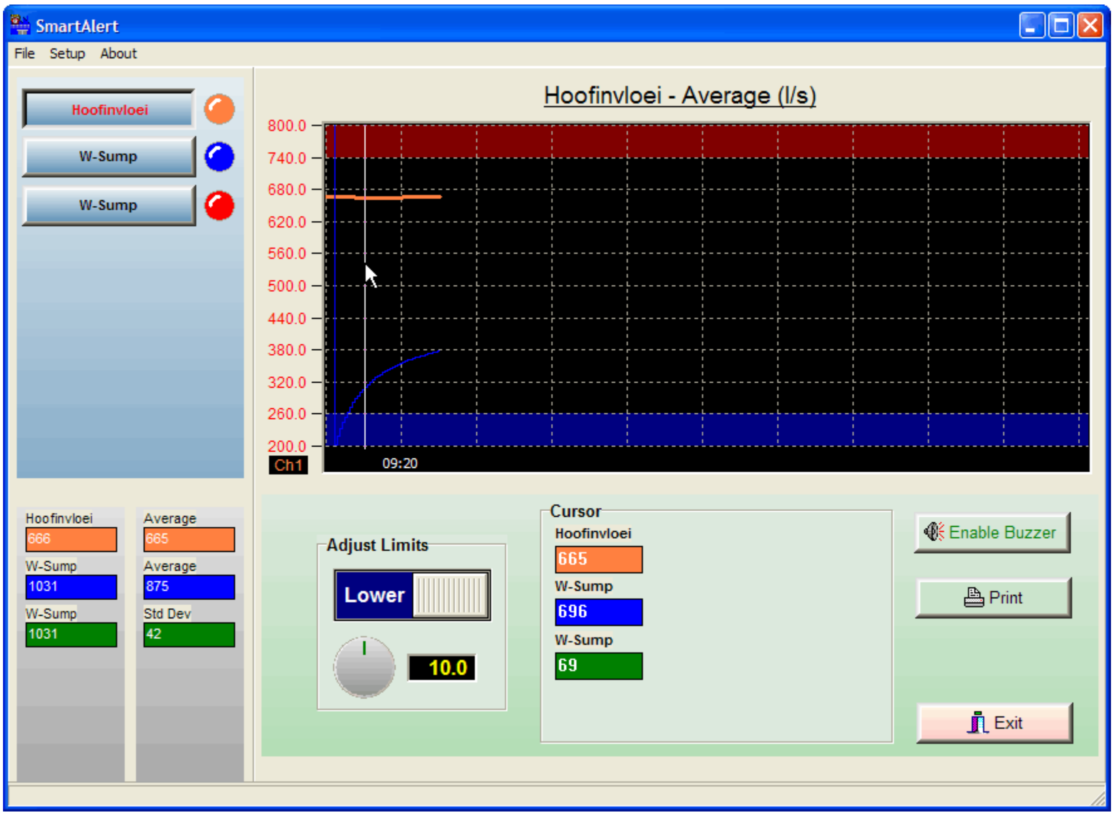

Figure 10: SmartAlert Display

One can select up to three "views" of each channel to base the alarm levels on; these can be "Average", "Standard deviation" or "Rate of change".

4 FIELD TRIALS AND COMMERCIALISATION

In order to ensure that the system meets the design targets, it was decided to trial the system in a number of different industries. To date the system has been implemented at six sites; two in automotive, two in waste water, one in generic hardware and one in food production. The reception of the system has invariably been enthusiastic by management; the need that was identified is a real one, having a system like Smartfactory has the potential to make a big difference to a company. Longer term adoption of the system has proven to be a problem though. Some examples of the problems are: the system was seen as a CSIR experiment and was treated as such, the system was provided free of charge and therefore lacked commitment from the plant, and in some cases the system not being integrated with the overall plant's systems meant that little attention was paid to the data.

The intention of the trials has been to test not only the system's reliability in both hardware and software, but also to verify whether the system applies to different industries and that it would indeed be easy to install and use. The point has been reached where a user can insert a single CD into a PC that will operate the system, and in around 30 minutes he will have a basic working system. An attempt was also made to verify the ease of installation of the hardware, and in two cases successful use has been made of an electrician with no formal instrumentation training to attain a working system. The system has proven itself from a reliability point of view; over about two years of operation only one single analog input on a SMARTi module has failed, and that event was probably due to a lightning strike.

As the system has now proven itself capable of performing the intended tasks, methods are now being explored to distribute it further into industry. One way is to license the technology to a commercial partner and talks have begun with a number of interested parties. Whether this is the right approach is still uncertain as the original intention was to make a low cost system available. The problem is that a potential agent and/or reseller will have to make a living also, thus upping the end costs for the users. However, these end costs should be still nowhere near to those that users would pay for similar imported alternatives. Other dissemination methods also have their problems; simply making the hardware and software “Open Source” might only attract some enthusiasts and would most probably not have the intended impact.

5 CONCLUSION

The Smartfactory solution presented in this paper meets the original design goals of the system. A well working solution has been created which is easy to implement and use and can be installed at a very low cost. The system can be installed on a trial basis using only a single module to verify operation, and can then be grown to a sizeable installation capturing plant-wide information.

6 RECOMMENDATIONS

It has always been the intention to eventually implement some form of control modules and software in addition to the monitoring function of the system. Pending further development funding this might occur in the future. To ensure a greater acceptance of the system the focus is currently on performing trials not aimed at proving the system technically, but proving the system commercially; proving what the benefits to a factory are once they have installed a system. This is not seen as a major obstacle, as a single observed event such as a gearbox heating up beyond normal operation and proper reaction to that event already has the potential of saving costs on machinery replacement and production losses which more than equal the installed cost of the Smartfactory system.

7 REFERENCES

[1] http://www.amts.co.za/ accessed 15 December 2009
[2] http://www.adroit.co.za/ accessed 15 December 2009
[3] http://en.wikipedia.org/wiki/Programmable_logic_controller accessed 16 April 2010
[4] http://en.wikipedia.org/wiki/SCADA accessed 16 April 2010
[5] http://en.wikipedia.org/wiki/EIA-485 accessed 16 April 2010
[6] http://www.ubuntu.com/ accessed 15 December 2009
[7] http://www.mysql.com/?bydis_dis_index=1 accessed 15 December 2009
[8] http://en.wikipedia.org/wiki/DIN_rail accessed 16 April 2010
[9] http://www.modbus.org/ accessed 16 April 2010