A Proposal for the
ATLAS MUON Spectrometer (AMS)
Detector Control System (DCS)

E.N. Gazis, M. Dris, and T.A. Filippas
NTU-Athens, Physics Department, Zografou Campus GR-157 80, Athens, Greece

and
H. Rahmani and E. Stiliaris
Institute of the Accelerating Systems and Applications (IASA) , P.O. Box 17214,
GR-10024 Athens, Greece

Abstract

An investigation is described for the definition of the requirements needed to monitor and control the ATLAS MUON Spectrometer (AMS). The existing expertise and the literature reference push to the adoption of the use, as much as, possible standard commercial products. A short term project based on the LabVIEW package by National Instruments is presented, developed for the needs of the CERN test beam facilities and MDT test facilities at the home institutes. A long term project is proposed to be developed for the overall AMS-DCS based on the EPICS environment, which can easily be integrated with the ATLAS-DCS and the other sub-detectors.

Contents:

1. Introduction and Overview

The ATLAS MUON Spectrometer - Detector Control System (AMS-DCS) is considered as a part of the whole ATLAS-DCS, where all the functions of controlling and monitoring the ATLAS hardware equipment are maintained, reporting and acting on changes in the status of the detector plus covering the relative safety aspects for the equipment and the users.

The ATLAS-DCS well distinguished from the data acquisition system (daq) of the experiment is strongly correlated with the daq and the trigger system [1.], interacting properly with these tasks.

The AMS-DCS is responsible for the control and monitor the Muon Spectrometer, the maintenance on the proper conditions of the chambers have been set-up. It communicates with the headquarters of the ATLAS-DCS reporting the status of the Muon detector, accepting commands for the hardware operation and transferring data back with reliability.

The major technical aspects needed to be controlled for the AMS normal commissioning are temperature, gas pressure, high voltage and magnetic field. Subsequent aspects like gas flow and mixture, anodes current and chamber alignment (planar, axial) should be controlled and monitored, too. The ATLAS safety services system (A3S) for hazardous conditions of the detector and its environment, complemented by a system of hardware interlocks, will be also extended to the Muon detector partition.

2. General Aspects and Characteristics

It is proposed in the next chapter a candidate prominent control system complying some of the following/forthcoming basic aspects and characteristics:

1. The detector control system, being a combination of a variety of different approaches, needs special care for the way of the integration at the operator's interface. For that reason general purpose controls applications must be provided which are reasonably easy even during initial operation.

2. The detector control system must accommodate and integrate the necessary possible control subsystems (modular type) and be able for possible new subsystems, appearing later, integration (open type).

3. The often called "standard model" of control systems **client-server configuration** is proposed to be adopted.[2.]

4. It is proposed the TCP/IP family as the communication protocol for the control environment to transfer and accept information data.

In the Server entity are considered all the devices like sensors and actors connected to a server program through a gam of electronics. In addition the program provides a service to the network. All requests to the device have to pass to the appropriate server program. Usually, the server resides in VME and communicates to the device via the "field bus". Several independent server processes provide support to the various devices of the ATLAS-DCS. The services vary from simple reads of device data, automatic controls of device functions and error handling up to complex calculations or processing of data from the several devices. Information from these services is available on the common ATLAS ethernet.

The client programs make use of those services displaying the status and the control of the detector. The client programs provide the screen graphical interactive display applications and screen drivers. The possible Client-Server configurations are illustrated in fig. 1.
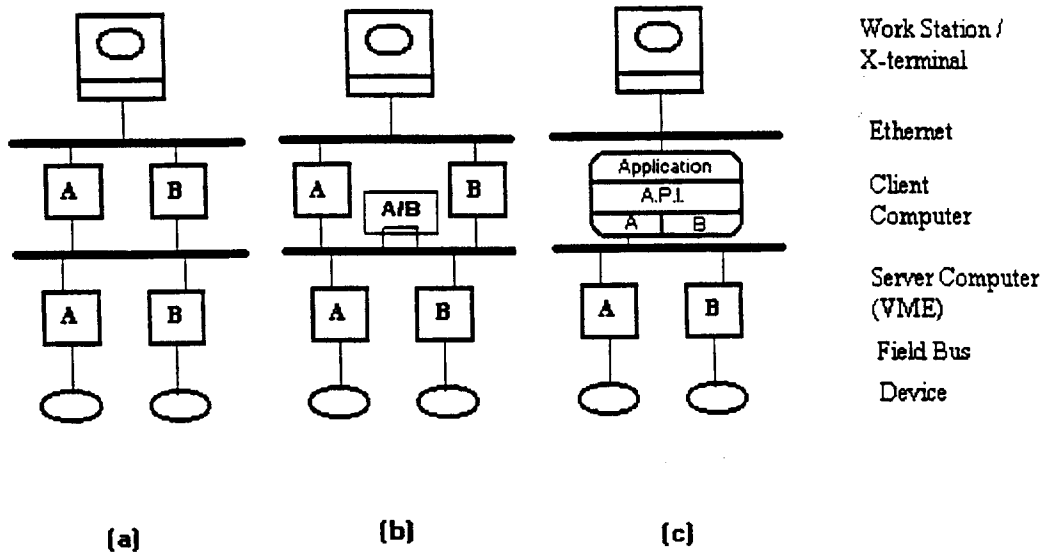
Fig. 1. (a) A simple Client-Server configuration with two paths to the monitor, (b) A configuration using "gateways" between one type of server and a different type client, (c) The application program interface (API) which integrates the approaches.

A first solution for integration of these protocols is to add, of what it is called, "gateways" for communication between the systems. The gateways are processes that talk protocols and can translate between them. The later step a common Application Program Interface (API) will be implemented.[3.] The API avoids the disadvantages of network traffic of the gateways. The API defines a set of standard calls with a set of commands. The API library detects from the commands which protocol to use.All the complexities of the network protocols are covered by the client programs.

The following candidate control systems may be proposed for the ATLAS-DCS development and integration.

1) The Experimental Physics and Industrial Control System (EPICS) developed at Los Alamos with the Channel Access (CA) protocol between clients and servers. [4.]

2) The Control Systems for the accelerating machines at CERN and DESY based on the Remote Procedure Call (RPC) communication. [5.]

3) Other systems based the commercial real-time kernel PSoS with the "classic protocol" on top of UDP developed at Fermilab. [6.]

3. Long Term Project
3.1 Features of the system EPICS

General

EPICS (Experimental Physics and Industrial Control System) is a distributed process control system built on a software communication bus, originally developed by Argonne National Laboratory and Los Alamos National Laboratory and currently used in many Accelerator Facilities in the USA and Europe (CEBAF, APS, TESLA/DESY, IASA). The system is scalable from a single test station with a low channel count to a large distributed network with thousand of channels. [4.]

EPICS provides all capabilities required by the Experimental Physics Community, being a modular constructed and extensible in order to handle new I/O devices and new technology for microprocessors.

Components

EPICS consists of a set of software components and tools with which the users can create a control system. EPICS execute high level applications and run a real-time kernel, provided by the VxWorks commercial package, for I/O by its (IOC) front-end computers as VME backplane.

Architecturally, it comprises three physical (hardware) layers and a set of software core, fig. 2.
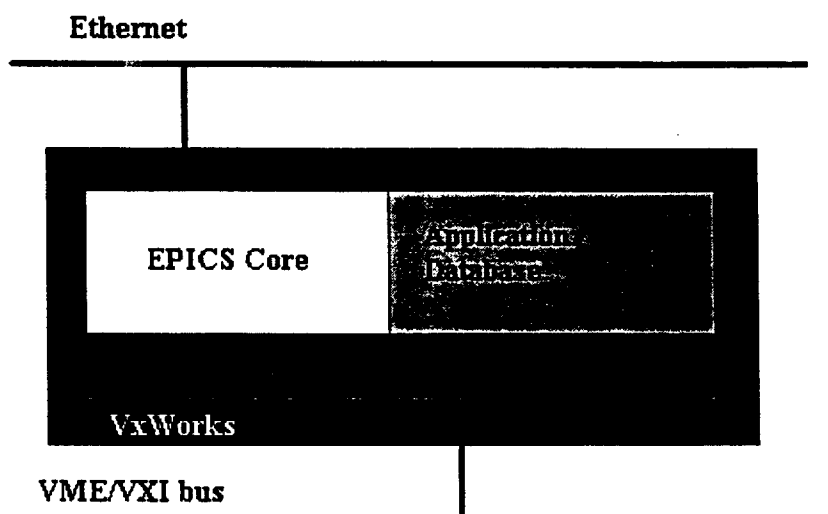
IOC Components Overview

Fig. 2. The IOC components overview

The basic components of the EPICS are:

- •OPI   Operator Interface. This is a UNIX based workstation which can run various EPICS tools.
- •IOC   Input Output Controller. This is VME/VXI based chassis containing a Motorola 68k processor [7.], various I/O modules, and VME modules that provide access to other I/O buses such as CAMAC.
- •LAN   Local Area Network. This is the communication network which allows the IOCs and OPIs to communicate. EPICS provides a softwarecomponent, *Channel Access*, which provides network transparent communication between a Channel Access Client and an arbitrary number of Channel Access Servers.

The physical front-end layer is built from VME crates,[8.] CPU and I/O boards while the physical back-end layer is implemented on popular workstations running Unix or VMS. Communication between them is achieved with a network layer, which is a combination of media (such as Ethernet, FDDI) supporting the TCP/IP internet protocol, where a scematic plot of the IOC components is presented in fig. 3.

EPICS Components

```
ETHERNET ◄───────────────────────────────────────────►
                              ▲
                              ▼
                    ┌─────────────────┐      ┌─────────────┐
                    │ Channel Access  │◄────►│  Sequencer  │
                    └─────────────────┘      └─────────────┘
                              ▲
                              ▼
┌──────────┐      ┌─────────────────┐      ┌─────────────┐
│ Scanners │◄────►│ DataBase Access │◄────►│  DATABASE   │
└──────────┘      └─────────────────┘      └─────────────┘
     ▲                        ▲                     │
     ▼                        ▼                     │
┌──────────┐                                        │
│Interrupts│                                        │
└──────────┘      ┌─────────────────────┐           │
     │            │ Record/Device/Driver│◄──────────┘
     │            │      SUPPORT        │
     │            └─────────────────────┘
     │                  ▲    ▲
     ▼                  ▼    ▼
═════════╪══════┌─────────────┐═══════════════════════
                │  VME/VXI    │
                └─────────────┘
```
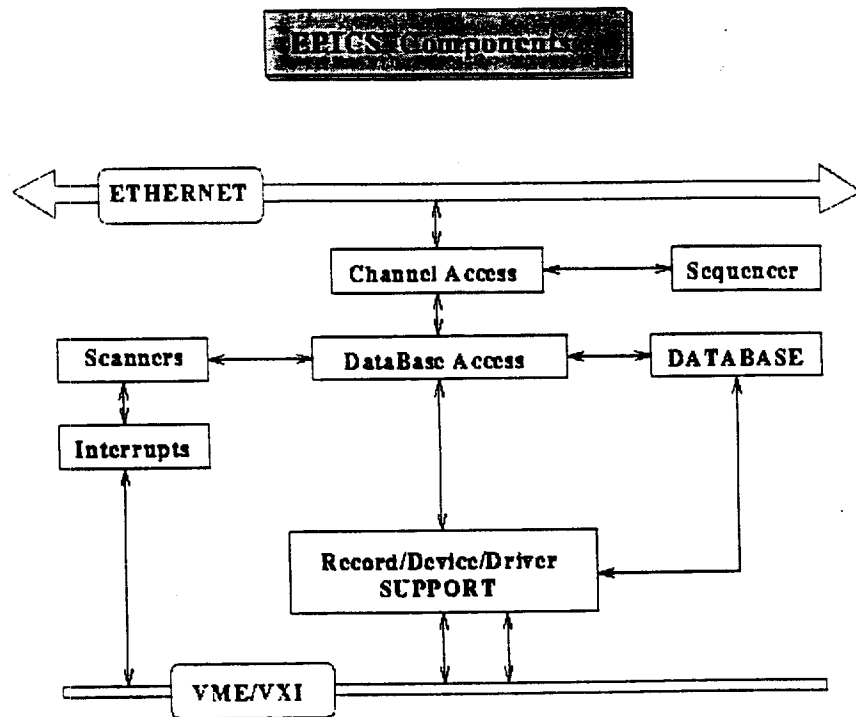
Fig. 3. The IOC Components

EPICS has no standard configuration. It is completely a user configured system, leaving a lot of freedom in the creation of the environment, in contrast to a complete, machine dedicated, control system. Some sets of I/O systems are preferred and recommended. Refinements and improvements of the EPICS software are based on users experience.

In order to reach a high level performance, the complexity of the various detector parts and their relationships have first to be well understood. Since the project priority is now shifted to the construction of the ATLAS DCS, all the hardware equipment of this detector part has to be described and interfaced to the control system.

Device Layer

The Driver Development at EPICS Device layer defines the different device drivers at the bottom layer of the software core. The necessary software support for all detector components and the other safety tasks has to be provided.

Database Support

The heart of the control system is the EPICS application database (described in terms of records), which defines all aspects of signal processing. The database records interface to the hardware through device support and driver support levels. EPICS is delivered with a set of generic device support routines, however, by their nature hardware interfaces tend to be application specific. Therefore, the successful implementation of the control system requires a lot of development steps, part of them being the object of this proposal.

Once the device is defined the database can be filled with definitions for all the I/O points. At this level different kind of actions can be designed and realized. Since the EPICS database is responsible for all aspects of signal processing, it must include all types of analog/binary input and output signals

necessary to setup and drive the hardware and to perform alarm handling. Low-level calculations and monitoring of several hardware parameters can be implemented here.

EPICS compiles or recompiles a portion of programs or databases. A common way of changing the database structure starts with a complete technical drawing of equipments using a Computer Aid Design package e.g. CapFast. It is possible to have other sets of database or any kind of CVS files created or modified by several spreadsheets or database software packages, as EXCEL, dBASE4 etc.(CVS files will be converted to the EPICS Short form by C- code source programs). The data of the mentioned files will then be converted to the EPICS database final form and finally to a binary form.

Any functional part and interfacing or translator of base is written in "C". Compilation and run of the sequences in C or partial recompilation of the source codes are performed by an special "make" file created by using the "GNUmake" package which is also an other commercial package.

Functionality

The functional layout of EPICS may be described by its Software Tools which are :

Alarm Handler : This tool is able to filter and display alarms, hierarchically.

Archiving Function : An archive server receives buffered data from the front-end via the Ethernet, which can be activated for off-line analysis.

Backup Restore Tool : Back up and restore tool addresses the problem of backing upand restoring sets of values in a dynamic database.

Configuration Management : a distributed database is used that is downloaded at start time to the front-end.

Data Base : This is a text based tool, allows the system designer to configure the database as required.

Devtest : Talks to IOCs through channel access routines.

Knob Manager : This tool allows the operator to connect the SunDials hardware knobs to process variables.

Motif based Editors Display Manager : MEDM provides a graphical user interface between the operator and the control system.

Network Communication : has a highly optimised network layer. It allows polling and monitors the client programs. Polling is used for single request of a client to a server to get data item. Monitors indicate the interest of a client of data update by the server. The server updates the client automatically on changes of the according value.

Server Process : a set of predefined records extensible is used to compose a server.

Sequencer Tool : This consists of two components: first is the run time sequencer (SEQ) and second is the state notation language(SNL) which provides a simple tool for programming sequential operation based on the state transition diagram concept without the complexity that task scheduling, event handling, and input-output programming involve. The sequencer on the client side allows the control of groups of devices and ordered sequences of procedures.

Application/Graphic Support : Once the database is defined and downloaded to the front-end, the data are available on the network by means of the Channel Access protocol.

High-Level Application Development

Once the database is defined and downloaded to the front-end, the data are available on the network by means of the Channel Access protocol. Some simple interactive tools are available for graphics, alarm handling and data manipulation. However, additional, more sophisticated applications

are needed to support the operation of the ATLAS-DCS complex. These applications will contain combined algorithms combining different database signals and will interact with other programs.

Software Organization and Maintenance

Finally, since all these activities are based on a UNIX development and operator interface environment, not necessarily homogeneous, another big priority has to be set in the maintenance and the optimal software distribution on this environment. Software tools and techniques widely used in the High Energy Physics Community can easily be adapted in order to optimize the creation and automatic distribution of machine-dependent files (libraries, executables) on the different platforms.

Additional programs can be written in 'C' or in any specific language based on the system states. This program can directly call the channel access(CA) interface to read and write data as desired.

Results expected

The main difference between the EPICS approach - which is a toolbox approach - and oth control systems for our detector is, that a toolbox leaves a lot of freedom to the one using it to cre his own environment according to his needs.

The primer goal is to obtain the basic functionality and performance necessary to commission an control the ATLAS detector. Since the robustness of the control system is given by the many years experience and development effort gone into the design of EPICS, a high quality can only be achieved by the careful description and integration of the hardware elements and their relationships. The implemenation of the high-level applications proposed here plays an important role in the detector performance and should therefore receive appropriate emphasis. It embodies the understanding of the complex mechanisms between the DAQ system the DCS and requires also an adequate knowledge of the trigger system

Finally, the modularity and the use of standards provided by EPICS allows vendor independence and gives the flexibility to follow future advances in technology. Standards also make it possible to share experience and to collaborate with other laboratories.

3.2 Hardware and Software

According to the C-S (client-server) model which we define to be the basis of our DCS, the front-end systems have to collect and compute the signals from the process. It is recommended nearly all front-end systems be based on VME. From here the field bus as well as the fast readout system is connected.

Front-ends

Since VME is an industry standard many different types of I/O are available. This enables us to choose from a variety of models and vendors and to pick the ones that fit the best as for the readout.If special requirements are necessary and/or the room on the VME-size boards is not sufficient, VXI (Vme eXtended Instrument) is an excellent addition to the VME system.

An EPICS minimum Hardware configuration is consisted by :

      a) A Unix workstation (SUN, DEC, HP, SGI, WINDOWS NT)

      b) Ethernet facility

      c) VME Crate (I/OC* and modules)**

      d) CAMAC (CAMAC controller and modules)**

      e) VME-CAMAC communication module, i.e. CBD 8210 by CES [9.], or CAMAC
         controller plus ethernet facility, i.e. VCC 2117 by CES. [9.]

    (*) The family of processors which is heavily used in VME are the Motorola 68000 CPUs, i.e.
      FIC8234 (by CES) [9.], MVME162, MVME167 (by Motorola)[10.]

    (**) A rich set of drivers for I/O boards are available among the HEP labs.

The necessary Software environment
  f) Stand alone EPICS
  j) GNUmake
  k) CapFast
  l) VxWorks (providing the real-time kernel)
  m) C compiler

VxWorks is a High-performance commercial real-time operating system and a powerful development environment for real-time applications, it can operate either stand-alone or networked with other system running VxWorks or UNIX.

The UNIX system is used for software development and the non-real·time components of high level applications. Cross-development package of VxWorks proved a powerful Unix compatible networking facilities.

Field-Bus

There are plenty of field buses could be proposed and implemented in our DCS. A brief description to some of them can give an idea :

SEDAC (Serial Data Acquisition)

The SEDAC is a proprietary field bus used at DESY. The origin of this bus is a stripped serial CAMAC. The advantage of SEDAC is the reliable transport of data over distances of more than 3 km the disadvantage is the low speed of about 15 kByte/s.

CAN (Computed Area Network) / CAN-developments

CAN is an ideal bus for fast and reliable applications. In addition single chip solutions for the CAN connection including a CPU and even on chip ADC's are available. The chips are reasonably cheap due to their application in cars and trucks. Besides the use of CAN has the advantage of a multimaste communication. The reliability of the CAN is limited according to the ratio between the bus speed and its length, i.e. at a length of 40 m the speed rewrites 1 MBit/s.

I/O subsystems are commercially available with several special developments (CAN-developments) like specialised front-end for temperature read-out, power supply control, interlock systems, etc.

SIEMENS-H1

The H1 bus is running on 802.3 Ethernet standard media like coax, fibre and twisted pair. It is devoted to perform as a backbone connection for distributed PLC (Programmable Logic Controller) systems.

Other similar buses proprietary of other forms are also available.

3.3 EPICS Minimal configuration at NTU

A minimal configuration of the system EPICS has been installed in our lab, where the VxWorks and EPICS are overlaid on a UNIX machine (SUN SPARK-5). The SUN workstation is connected via ethernet with the VME fast intelligent controller FIC8234/Motorola68040/25MHz. The communication, commands and data transfer between VME and CAMAC crates is realised through the CBD8210 module and the CCA2 controller keeping on the camac branch highway. A block diagram is appearing on fig. 4.

SUN - SPARK

**VxWorks /
EPICS**

VME Crate

**FIC
8234**   **CBD
8210**

**CAMAC Branch
Highway**
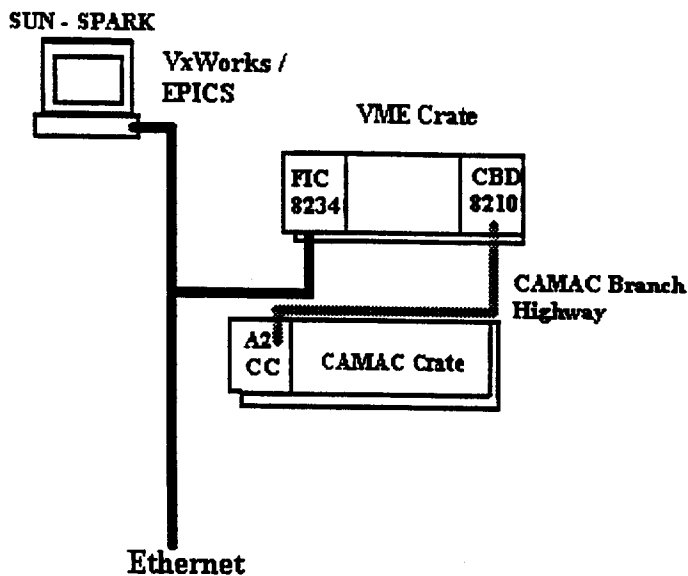
**A2
CC**   **CAMAC Crate**

Ethernet

Fig. 4. The minimal EPICS hardware configuration installed at NTU

The installation debugging has been finished and control loops are on routine exercise with a large number of available modules (mostly DACs, ADCs, and Motor Steppers).

4. Short Term Project
4.1 Features of the LabVIEW system

LabVIEW is graphical programming environment developed by National Instruments Co [11.]. LabVIEW runs on the Macintosh, Windows, Sun and HPUX environments, with programs being essentially transportable from one platform to another.

In the LabVIEW framework the Virtual Instrument (VI) is created according to the object to be controlled. The VI may be composed in subparts by other VIs available by the LabVIEW library or be written from the beginning with the help of the rich variety of graphical tools in the LabVIEW environment.

A VI program is consisted by two distinct parts:

1. The Front Panel, which displays the operator's working space, with buttons, switches, indicating lights, alarms. counters, plots etc.

2. The Diagram, which contains a kind of a flow chart. This chart connects the logic units of the instrument, gives the proper priorities during the program execution and presents the controlling results of the instrument to the front panel.

LabVIEW offers a user friendly interface program that makes the procedure relatively easy to develop the detector control system for temperature, gas pressure and high voltage. The extensive library of sub-programs (sub-VI's) of data acquisition, interface programs (e.g. GPIB functions) and analysis programs (e.g. curve fitting or advanced mathematical functions such as integration) kes the programming equally comfortable.

In addition the LabVIEW offers TCP/IP capabilities, linking specialised codes, if needed a: he ability to create stand-alone programs for the working platform. A main disadvantage is .hat LabVIEW is not a real-time environment.

LabVIEW is ideal system for fast development of relatively small, self-contained systems, where real-time reactions are not necessary.

4.2 LabVIEW Minimal configuration at NTU

A minimal configuration of the LabVIEW system has been installed in our lab, where source code has been loaded to a MacIntosh-centris 630. The Macintosh is connected via the VIC-bus interface card MAC-7212 with the VIC-8250 a one-slot VMV master/slave VME interface offering a transparent interconnection to the VMV bus. The VIC-bus is continued to the CAMAC crate with the VCC-2117 intelligent camac crate controller through which the camac modules ADCs, TDCs, etc., are controlled. A block diagram is appearing on fig. 5.
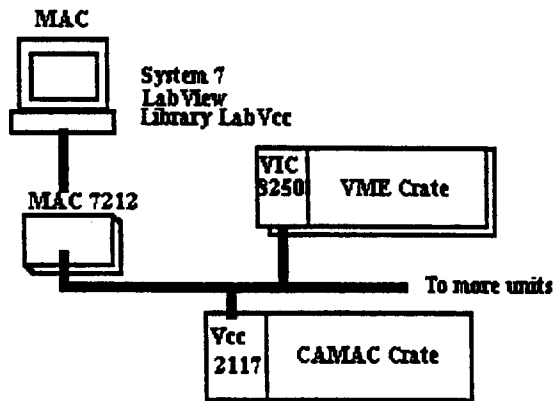

Fig. 5. The LabVIEW hardware configuration installed at NTU.

4.3 MDT.VI by the LabVIEW

The MDT.VI (preliminary) developed in our lab monitors the temperature of the detector system, the pressure of the gas mixture circulated into the chambers, the High Voltage applied on the anode wires of the drift tubes and the magnetic B-field, which is applied in the experimental site.

The Panel window provides the menu of monitoring the above selection tasks, while the selected parameter is plotted in a histogram. Upper and lower limits of the controlled values are activated when they are bypassed. The MDT.VI front panel is illustrated at the fig. 6.
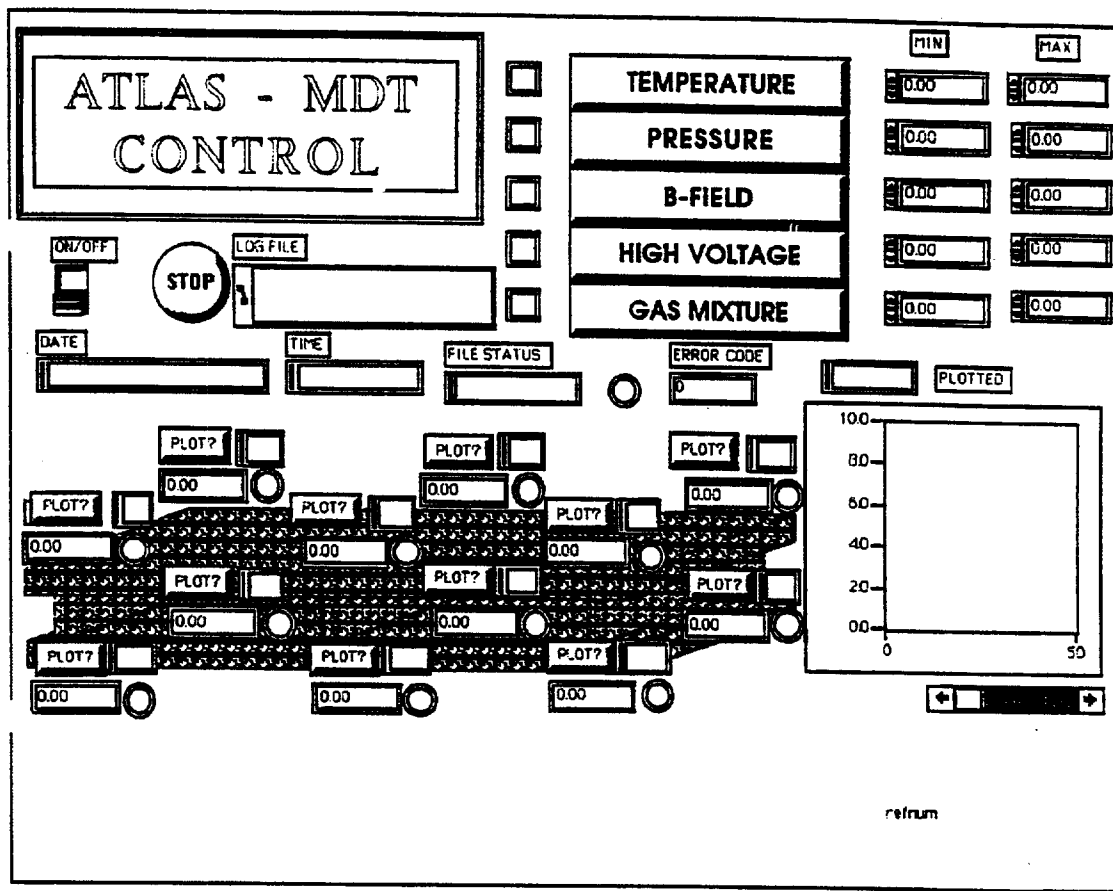
Fig. 6. The MDT.VI front panel (preliminary).

The block diagram of the MDT.VI is presented in fig. 7, where the logic of control the temperature, gas-pressure, H.V. and B-filed is indicated for each MDT chamber.

The same philosophy of the control could hold for the other chambers of the AMS, i.e. RPCs, CSCs and TGCs.   In addition, any possible further special feature could be incorporated in the VI that matches properly the sub-detector.
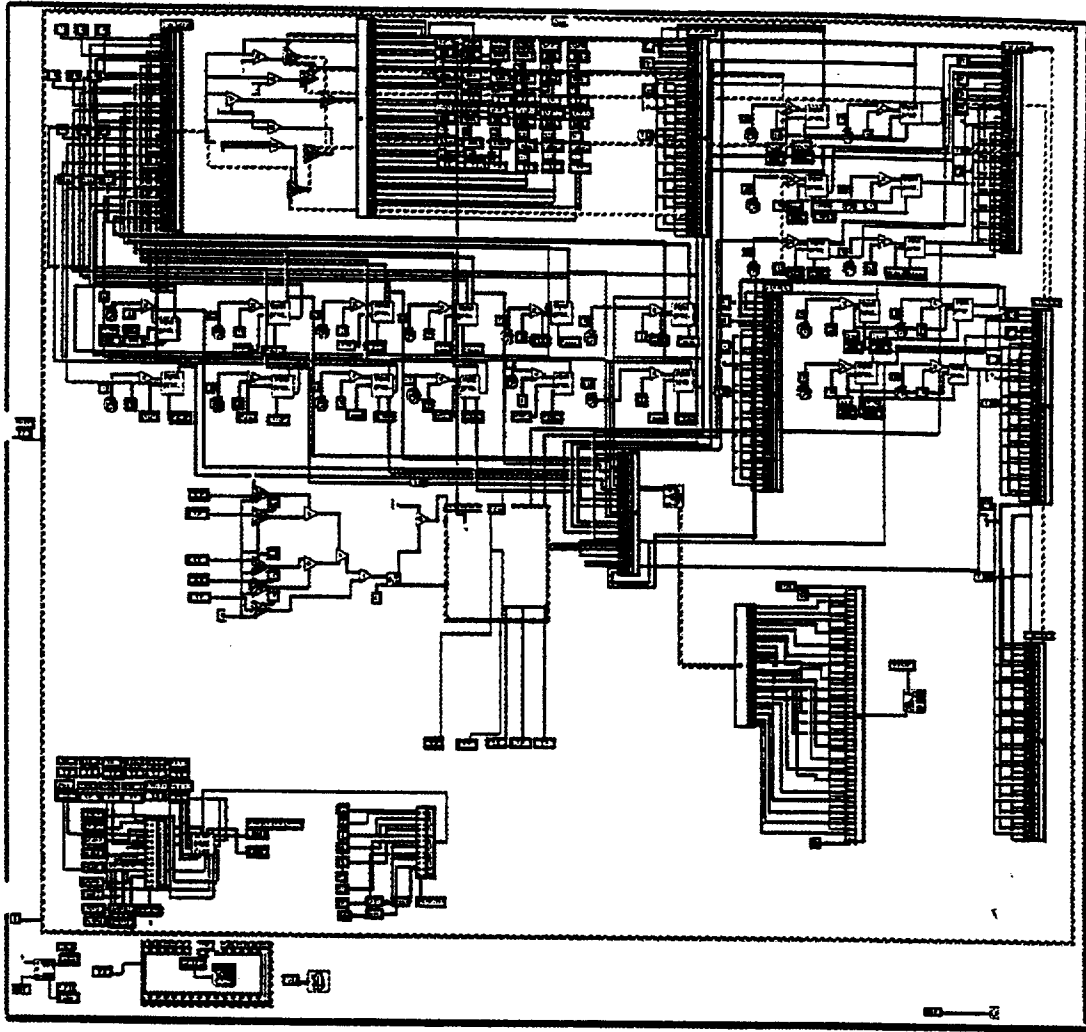
Fig. 7. The MDT.VI block diagram (preliminary).

5. Conclusions

This note with the proposed ATLAS DCS especially with some features suitable for the ATLAS MUON spectrometer has a pilot role trying to describe as much as possible the basic principles of the detector control system.

The finally adopted DCS has to be significantly developed further as higher and higher level systems will be added by the detector experts. In the mean time, the structure of the software tools and in particular our experience with the hardware, in order to make the proper choice of that, will be our big challenge for the recent next years.

In addition, the data acquisition system developed under analogous conditions has to interface smoothly with the DCS.

6. References

[1.] H. Burkhard, User Requirements Design for ATLAS DCS.

[2.] Client - Server Architecure, by A. Berson, McGraw-Hill, 1992.

[3.] F. Di Maio et al., Towards a Common Object Model and API for Accelerator Controls, Proc. Int. Conf. Accelerator and Large Exper. Phys. Control Systems (ICALEPCS), Berlin, Oct. 18-22, 1993.

[4.] L. Dalesio et al., The EPICS Architecture, Proc. Int. Conf. Accelerator and Large Exper. Phys. Control Systems (ICALEPCS), Berlin, Oct. 18-22, 1993.
W. McDowell, et. al, Standards in the Design of the Advanced Photon Source Control System, Proc. Int. Conf. Accelerator and Large Exper. Phys. Control Systems (ICALEPCS), KEK, Tsukuba, Japan, Nov. 11-15, 1991.

[5.] T.J. Berners-Lee, Proc. 5th Conf. on Real-Time Computer Applications in Nuclear, Particle and Plasma Physics (RT87, San Francisco, May 1987), ed. R.T. Kouzes, IEEE Trans. Nucl. Sci.34 (1987) 1050.
T.J. Berners-Lee, RPC User Manual, Version 2.4.0, CERN/DD/OC Writeup, July 1989.

[6.] TESLA/DESY Control Design Report, 1994

[7.] Motorola Inc., MC68000-family Microprocessor Programming Manual, Austin, Texas.

[8.] "The VME bus specification" ANSI/IEEE 1014-1987, IEC 821.

[9.] CBD8210, Camac Branch Driver, User's manual v1.1,
VCC2117, Intelligent Camac Crate Controller, User's manual v1.1,
FIC8234, Single-board workstation, User's Manual, v1.2
By Creative Electronics Systems Co., Geneva, Switzerland.

[10.] MVME162 / MVME176, Single-board computer, User's Manual, Motorola Inc.

[11.] LabVIEW v.4, National Instruments, Austin Texas.