United Kingdom Atomic Energy Authority

UKAEA FUS 523

S. Shibaev, S.J. Manhood, G. McArdle

MAST real-time display

MAST real-time display

S. Shibaev, S.J. Manhood, G. McArdle

EURATOM/UKAEA Fusion Association, Culham Science Centre, OX14 3DB, Abingdon, UK

UKAEA FUS 523

EURATOM/UKAEA Fusion

MAST real-time display

S. Shibaev, S.J. Manhood and G. McArdle

June 2005

EURATOM/UKAEA Fusion Association

Culham Science Centre
Abingdon
Oxfordshire
OX14 3DB
United Kingdom

Telephone: +44 1235 820220
Facsimile: +44 1235 463209

MAST real-time display

S. Shibaev, S.J. Manhood, G. McArdle

EURATOM/UKAEA Fusion Association,

Culham Science Centre, Abingdon, Oxfordshire OX14 3DB, UK

Contents

Introduction

The collection and processing of mainstream data on Mega-Amp Spherical Tokamak (MAST) [1] currently takes 3-6 minutes which is a considerable part of the inter-shot interval. The physicists and engineers performing experiments needed a tool for viewing the key operational data and plasma parameters immediately after the shot. In view of the future MAST upgrade and longer shots we decided to implement this tool straight as a real-time one. The displaying of data in real-time eliminates any problems with future extensions and makes possible to implement other useful functions.

Another similar problem is to present the same data to the MAST team outside the control room, and to remote participants. That could be realised as a dynamic web page. And a natural way is to solve both problems in one system.

There were two combined tasks – to collect signals from several remote sources and visually present them in the MAST control room in real-time and on a web page which should be updated immediately after the shot. Most of the signals collected need simple data processing before displaying, such as zero level subtraction and/or multiplication by some scaling factor. The set of signals to be displayed together with their scaling and position on screen can vary in different experiments. That requires a user-friendly graphical interface providing full configuration without any programming.

A natural requirement is to implement the real-time display as simply as possible, with minimum use of new hardware and minimum software development.

Choice of architecture

There are, in principle, two ways to implement data collection from remote sources. The first solution uses a real-time data acquisition system – a set of workstations connected by some network – Ethernet, ATM, *etc*. Each workstation collects and processes a group of signals in real-time and outputs to the network a reduced sub-set of signals. Such data acquisition systems are realised on several fusion facilities [2, 3]. But this way is very expensive both in hardware installation and software development, and it is not justified for the requirements of MAST.

We chose another way – to collect a set of raw signals in one computer and digitise them by a single ADC. This solution is optimal for our task because the number of signals to be displayed on one screen is naturally limited, and it needs only one control program or set of threads running on the same computer without the use of any network.

The signals from different remote sources must be isolated, and the use of long electric cables is inappropriate for a facility with high alternating magnetic field. That leaves only the choice to use fibre–optic analogue transmission lines. Hence, the initial design was as follows: a PC in the MAST control room with one multi-channel ADC; all signals should be connected by fibre-optic analogue transmission lines; the control program collects data and displays them on the local PC monitor (projector).

There is one logical defect in this scheme – the analogue transmission line is in principle a pair of ADC and DAC because data are transferred through the fibre cable in digital form. So, there are two spare data conversions: ADC – DAC – ADC. Unfortunately, the first device requiring only one data conversion appeared on the market when the MAST real-time display had been already installed. This type of device, “remote ADC”, consists of a multi-channel ADC with a fibre-optic output and a PCI or USB card inputting data directly into the PC without spare conversions. An example is described in [4]; this approach could be used for many tasks when analogue data transfers are needed.

Hardware implementation

In fact, the original design has been changed radically during implementation. The main reason was that more than half of the required signals, mainly magnetic signals, had already been marshalled into one cubicle for the plasma control system. To reduce the number of analogue transmission lines required we decided to install the computer with the ADC in the same cubicle in the MAST area and connect the available signals straight to the ADC. Other signals are connected by analogue transmission lines. This solution involves a second computer in the control room – that computer serves as a remote display for the main computer. It shows the main computer desktop on the projector screen in the MAST control room. To avoid interference with the MAST network during shot cycle (the busiest period) both computers are connected by a private network line. This scheme uses the "Remote Desktop" facility implemented in Windows XP.

We use an ADC (PCI-DAS 6402/16) from Measurement Computing [5] with input multiplexing – not simultaneous acquisition. The ADC has 32 differential or 64 single-ended inputs. The ADC resolution is 16 bit, with a maximum acquisition rate of 200 kHz. These characteristics are quite satisfactory for our task. At present we use 20 differential channels, giving 10 kHz sampling rate per channel. The sensitivity of each channel can be configured independently with uni- or bipolar ranges from 1.25 V to 10 V.

To connect remote signals we use fibre-optic analogue transmission lines – both made in-house of Culham (1 kHz bandwidth) and made by AA lab [4] (10 kHz bandwidth).

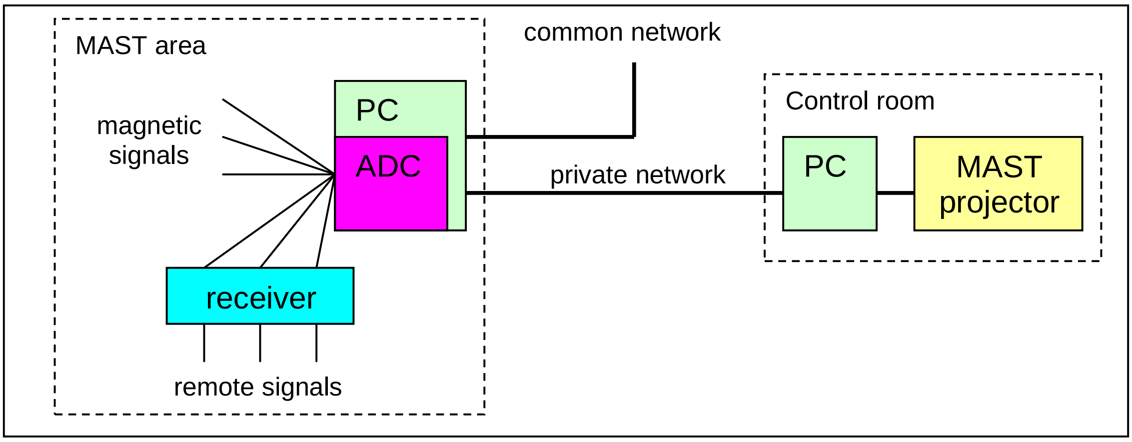


Fig. 1
The scheme of hardware implementation.

Control and display program

The control and display program has been developed in C++ for Windows; it uses only native Windows functions. The graphical user interface includes the display itself and extensive configuration tools.

Program logic

There are two logical parts of the program – the data acquisition and the data presentation. Hence, the program configuration consists of two parts – the program (and ADC) configuration and the "Layout" – a description of the screen, a selection of signals to display, *etc*. The data processing also consists of two stages. First, the raw data are scaled with calibration factors to produce physical values and the initial zero levels are subtracted if required. The data thus obtained are stored in the local file which is later copied to the central archive. The second stage of data processing is described in the layout – it can include sum or multiplication of two signals, the zero level can be subtracted in a defined period, and finally the resulting signal is scaled to fit the designed plot.

Splitting of the configuration and the data processing into two parts makes it possible to view the same data in different layouts, and those layouts can be prepared beforehand. Hence, the layout is implemented as an external text file. The program incorporates a graphical layout editor. The prepared layout can be quickly loaded and used for data viewing.

There are two program modes – running and stopped. The program configuration and the layout editor are accessible in "stopped" mode. In this mode the program can be used to view the data saved in previous shots.

When the program is started it connects to the central MAST message server. In "running" mode the program waits for the next MAST shot (the central "Trigger" state). On receiving that state the program clears the screen and creates a data acquisition and processing thread. The thread waits for the hardware trigger – first data available and starts data processing and displaying. The data are transferred from the ADC to the PC memory via the DMA in portions; the minimum portion is the ADC RAM buffer – 8k. That introduces a delay between data digitising and processing – up to 20 ms in the present configuration. The delay can be made much less using a faster (and more expensive) ADC, but it is not very significant for our task – the real-time display is currently used only for data viewing and the "Remote Desktop" noticeably slows down the drawing on screen. That doesn't influence the data processing and possible generation of derived events.

Main program window

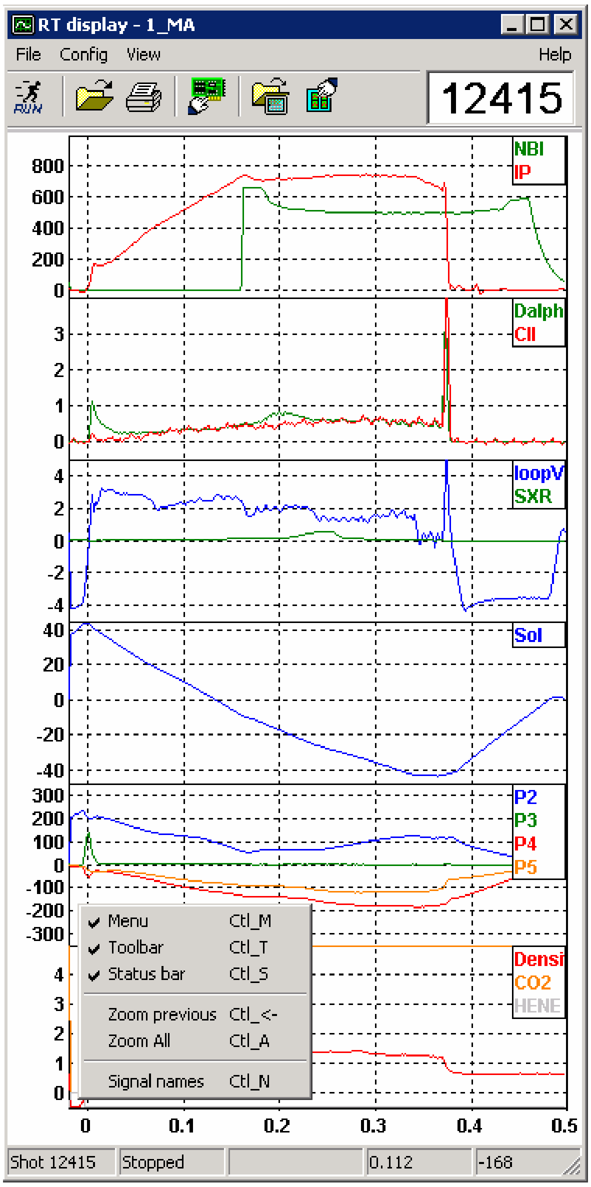


Fig. 2
The main program window.

The program GUI consists of the main window and several dialog windows for the program and ADC configuration and for the layout editor. The main window (Fig. 2) contains the data display and three bars – a menu, a toolbar, and a status bar. All three bars can be switched on/off in any time.

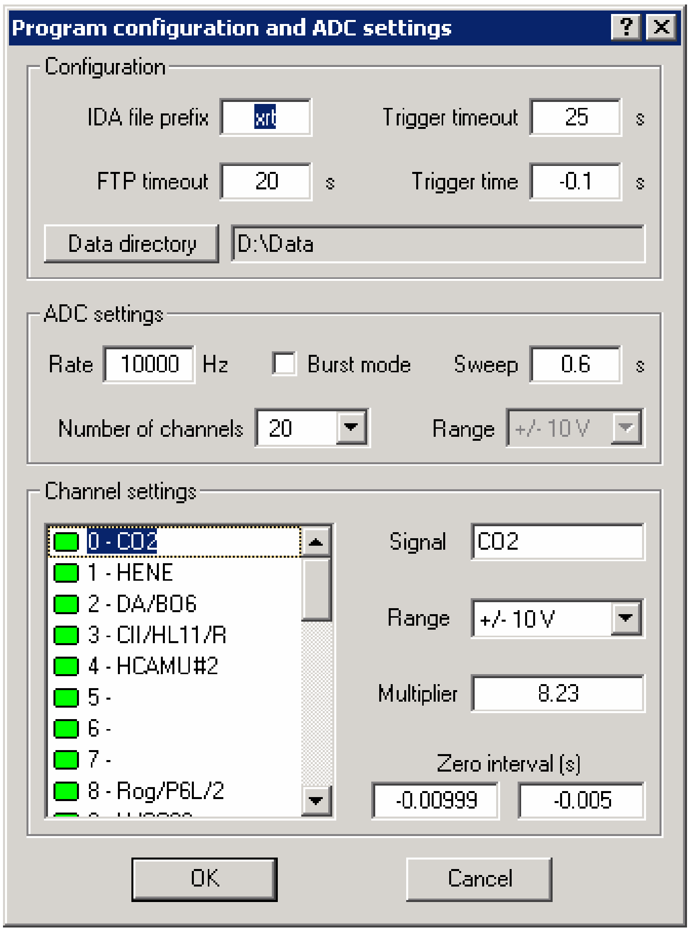
The toolbar contains the most often used functions (buttons) and a window showing the shot number of latest acquired or loaded data. The "Run" button switches the program into "running" mode, and the button changes into "Stop". There are also buttons: for loading data of previous shots, for printing of current view, for the program and ADC configuration, for loading a new layout, and for calling the layout editor. The load, configuration, and layout buttons are greyed (inaccessible) in the "running" mode. All buttons are supplied with a ToolTip help.

The program menu contains all functions, and it is useful only for initial familiarisation with the program because most functions can be called by toolbar buttons or by shortcut keys. The "view" sub-menu is duplicated as a shortcut menu called by the right mouse button (shown in Fig. 2).

The status bar has five sections, they show: the shot number; the program mode ("Running" or "Stopped"); the current program state ("Acquiring data", "Transferring data", *etc*). The last two sections show the mouse pointer position on the plot as a time and value pair.

The data display consists of several plots with common time axis. Everything on the screen is configurable in the layout – number and position of plots and signals, trace and background colours, font, *etc*. The display supports two zoom modes – a time window and a single plot. The click and drag of the left mouse button selects the time window. The double click of left mouse button zooms one plot into the whole screen. All manipulations with the display are safe for operations. In “running” mode the screen is cleared and reset just before the MAST shot.

Program and ADC configuration



The program and ADC configuration dialog (Fig.3) contains data acquisition and processing (first stage) settings. The “Configuration” panel contains the data file and MAST interface settings; this is practically the constant part of the configuration.

The “ADC settings” panel lets change the main ADC parameters – the acquisition rate (here is the rate for each channel), the acquisition duration, the number of channels, etc. The ADC supports any sequence of input channels, but to simplify the user interface the program supports the acquisition of only first of 32 channels and only in the channel number order. The ADC range is common for all channels in a “burst” mode and it can be set separately for each channel otherwise. Actually the “burst” mode, with fixed 5 μs interval between channels, makes sense for this card if number of channels is less than 20, and we actually don't use it.

The "Channel settings" panel contains the ADC and primary data processing settings for each channel. The program uses and saves only signals with a not-empty name and only from active channels. The program actually saves the settings for all 32 channels, so the number of channels can be safely changed, for example, if a faster acquisition on first channels is required. Each signal can be multiplied by the value entered in the "Multiplier" input – a negative value inverts the signal. The signal zero value is calculated during the interval between two times entered in the "Zero interval" inputs and subtracted from the rest of the signal in real time.

Layout editor

The display configuration is stored in a separate file. That gives an opportunity to prepare a set of configurations for different MAST scenarios, and the stored data can be displayed in different views.

The layout contains several common settings and a tree structure of plots and traces. The screen contains a number of plots; each plot contains several traces. The trace presents one signal or a combination of two signals. The layout editor follows that structure – the main window contains common settings and buttons calling the plot configuration. The plot configuration in its turn contains plot settings and a button calling trace configuration.

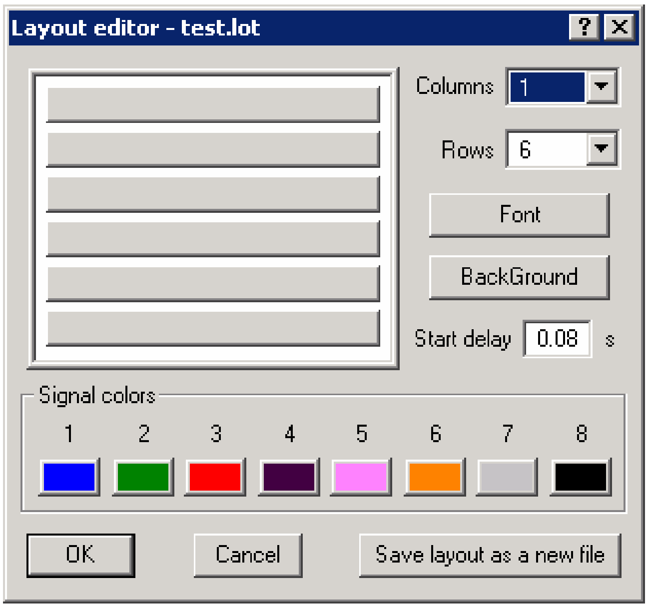


Fig. 4
The main window of layout editor.

The editor's main window (Fig. 4) has a schematic representation of the display screen in the top left corner. It shows selected number of plots; they are arranged in a grid – up to 4 columns

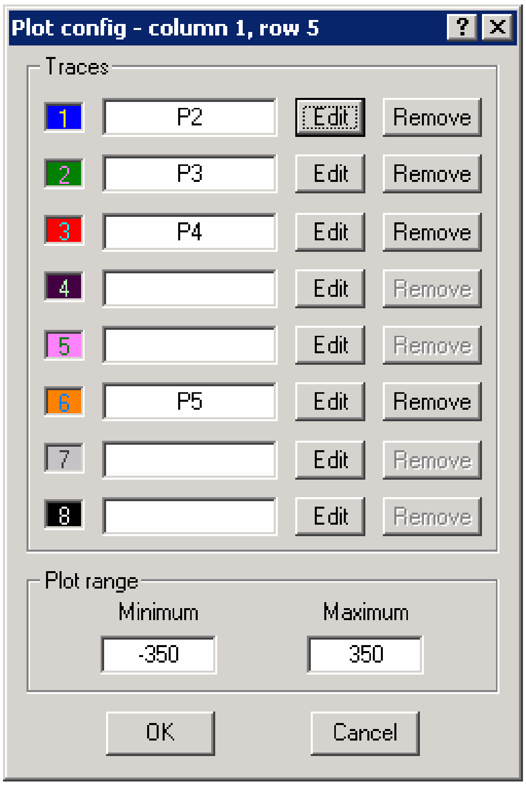


Fig. 5
The plot configuration.

and up to 8 rows. All plots have common time axis and rows in the column are packed densely. Each plot is shown as a button. The button calls the plot configuration dialog (Fig. 5).

There are common settings for all plots – the “Font” and “Background” buttons. These buttons call corresponding dialog boxes.

The data acquisition starts on receiving the MAST trigger (usually at time –0.1 s). The real-time display can start later; this time can be set in the “Start delay” input.

Each plot can contain up to 8 traces. The “Signal colours” buttons let change the colour associated with the trace number. These settings are common for all plots, but any signal can be selected into any trace, that is it can be drawn in any colour from chosen set of colours. Each colour button calls the colour selection dialog.

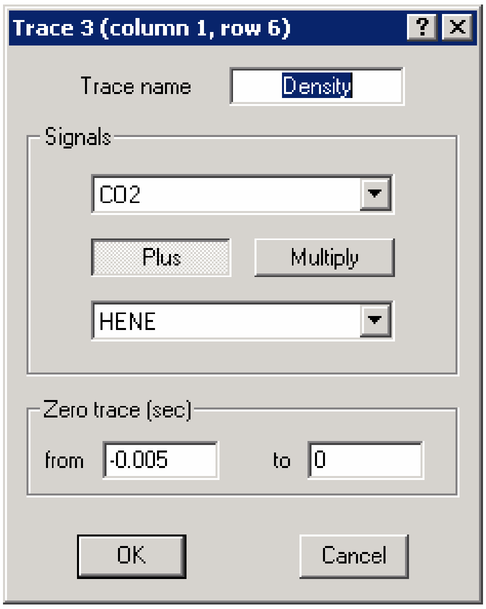


Fig. 6
The trace configuration.

The plot configuration (Fig.5) dialog lets select set of traces in the plot. The “Traces” panel contains 8 rows. Each row has a trace number (colour), a trace name, and two buttons. The “Edit” button calls the trace configuration dialog (Fig.6). The “Remove” button deletes the trace.

The range of values is fixed for each plot. This is the only possible solution for real-time display, and it is actually needed for quick estimation of the signals displayed.

The trace configuration dialog (Fig.6) lets select one signal, or sum or multiplication of two signals. Fig.6 shows an example of the trace configuration – the real-time plasma density signal. This trace is produced as a difference of signals from two interferometers – CO2 and HENE. HENE signal is inverted in the ADC configuration by a negative multiplier.

The resulting signal can be also set to zero in the specified interval. The value obtained as an average in this interval is subtracted from all following values.

Web interface

A quick presentation of the shot data is necessary for remote participation and it is very useful for the MAST team as a first indicator what's going on in operations. The most convenient solution for both tasks is a web page showing the latest screen of the real-time display. The web page must be a dynamic one; it must be updated when the MAST shot finishes. We chose the simplest implementation – the web page shows an image generated by the control program of the real-time display. The dynamic web page is implemented using Java technology – client side only.

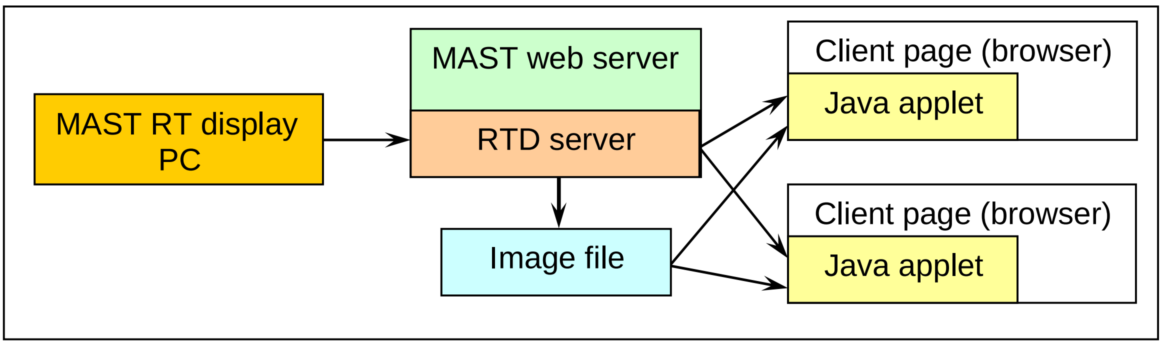


Fig. 7
The scheme of web interface.

Fig.7 shows the scheme adopted for the web page of the real-time display. When the MAST shot finishes (the real-time acquisition ends) the control program creates an image file which shows the data plots in the current layout. This is a copy of the screen drawn on another canvas; it is actually drawn by the printing function. Then the program sends the image and the shot number to the RTD server.

The specially designed RTD server has two network interfaces. It receives the image file from the real-time display through the MAST internal network and then dispatches an event to all connected clients (browsers) into an external network. The RTD server runs on the same computer as the MAST web server. The web page contains a Java applet. The applet shows the image on the browser's screen. When the browser downloads the page the applet connects to the RTD server and waits for the event, the next MAST shot. On receiving the event the applet downloads and shows the image.

This scheme proves to be very flexible and it can be easily extended for dispatching other display-related events or data generated in the real-time display. For example, it can output information of the value and duration of the plasma current or of the NBI pulse.

Summary

A real-time display for MAST data has been developed. The control program acquires up to 32 signals, processes and shows the data in the control room. The display has an extensive graphical user interface – everything on screen is configurable. The screen configuration is stored in a separate 'layout' file. Thus one can prepare a set of layouts for different MAST scenarios, and view the same data in different layouts. The control program includes a graphical layout editor.

A web interface for the real-time display has been developed. The interface provides a dynamic web page. The page shows a copy of the real-time screen and updates it just after the MAST shot.

The real-time display has been working faultlessly since 2002. The web interface was implemented last year.

The system is used in MAST operations. It is ready for use in the remote participation.

Acknowledgements

This work was funded jointly by the United Kingdom Engineering and Physical Sciences Research Council and by the European Communities under the contract of Association between EURATOM and UKAEA. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

References

[1] B. Lloyd *et al*., Nucl. Fusion **43** (2003), 1665.

[2] R. Felton *et al*., Proceedings of the 23rd SOFT Conference, (Venice, 20-24 Sep 2004).

[3] Y. Buravand, B. Guillerminet, Fusion Eng. Des. **71** (2004), 111.

[4] Fibre optic products, http://www.lab-systems.com.

[5] PCI-DAS6402/16, http://www.measurementcomputing.com.