A REAL-TIME DATA LOGGER FOR THE MICE SUPERCONDUCTING MAGNETS

J.T.G. Wilson, STFC Daresbury Laboratory, Warrington, UK

Abstract

The Muon Ionisation Cooling Experiment (MICE) being constructed at STFC's Rutherford Appleton Laboratory will allow scientists to gain working experience of the design, construction and operation of a muon cooling channel. Among the key components are a number of superconducting solenoid and focus coil magnets specially designed for the MICE project and built by industrial partners.

During testing it became apparent that fast, real-time logging of magnet performance before, during and after a quench was required to diagnose unexpected magnet behaviour. To this end a National Instruments Compact RIO (cRIO) data logger system was created, so that it was possible to see how the quench propagates through the magnet. The software was written in Real-Time LabVIEW and makes full use of the cRIO built-in FPGA to obtain synchronised, multi-channel data logging at rates of up to 10 kHz.

This paper will explain the design and capabilities of the created system, how it has helped to better understand the internal behaviour of the magnets during a quench and additional development to allow simultaneous logging of multiple magnets and integration into the existing EPICS control system.

MICE

The Muon Ionisation Cooling Experiment (MICE) is an international collaboration of particle and accelerator physicists from Europe, the US and Japan. It seeks to design, build and operate a muon ionisation cooling channel, which given the consequence of the short muon lifetime that makes traditional cooling techniques inappropriate, is an essential technology for the design of a muon collider or neutrino factory[1].

The MICE cooling channel is of the same design as the cells proposed for the neutrino factory and consists of 3 absorber coil modules with low density absorbers inside a focusing magnetic field and 2 RF-coupling coil modules. It is being built on a dedicated muon beam from the ISIS accelerator at Rutherford Appleton Laboratory.

MAGNET QUENCHES

Upon testing of the first Focus Coil magnet a series of unexpected magnet quenches were occurring meaning that the magnets were not able to reach the power levels specified by the design requirements.

There was already a quench detection system installed, however, this was only designed for machine protection, shutting down the magnets in the event of a quench and was thus not designed to monitor all of the individual coil power levels. This also led to doubts as to whether there were actually magnet quenches occurring or if the quench detection system was not functioning correctly. To be able to properly address the unexpected quenches and prove the integrity of the quench detection system it was decided that a further diagnostic tool was required to monitor the power levels on each of the coils of the magnet so that the starting point of a quench could be determined and it's propagation through the rest of the magnet analysed.

The proposed solution to this was to create a standalone logging system that could capture magnet performance data before, during and after a quench. Because of the unexpected and unpredictable nature of the magnet quenches (testing could be running for hours before experiencing a quench) a system that simply logged the values from the magnets as soon as they were turned on would create far too much unnecessary data. Similarly, a system that started logging only once it had received the signal from the quench detector would miss vital information because data showing the quench starting to build up on the coils was needed for diagnosis of the fault. The solution to this was to have a system with a 'rolling capture window'. This 'window', or buffer, would temporarily save the data (i.e. in RAM) and once the buffer was full would start to overwrite the oldest data in the buffer with the newest. This would allow for the system to have already captured and be temporarily holding the data showing a voltage differential building up to a quench which, after receiving a signal from the quench detection system, it could amend with the data during and after a quench.

THE DATA LOGGER SYSTEM

Hardware

CompactRIO is a reconfigurable embedded control and acquisition system. The CompactRIO system's rugged hardware architecture includes I/O modules, a reconfigurable FPGA, and an embedded controller [2].

For this application it was decided that NI 9222 4 channel C-series modules [3] were needed, despite the greater cost than other similar analog cards, to achieve the necessary sample rates. These cards can provide up to 500 kS/s per channel at a 16-bit resolution. These were coupled with the NI 9103 chassis [4] which provides 4 C-series module slots and the NI 9012 controller [5].

Additional external electronics were needed to reduce the input voltages down to the +/- 10 Volts that the NI

ISBN 978-3-95450-146-5

9222 cards can handle. All of this was then packaged up into a 4U rack mountable crate.

Software

The cRIO is designed to be programmed using LabVIEW. LabVIEW is a graphical programming language created by National Instruments and is designed to make programming and configuration of their hardware quicker and simpler for the user.

Specific modules of NI LabVIEW were needed for this project, the LabVIEW real-time and LabVIEW FPGA modules which allow for configuring the on-board FPGA chassis using the same tools and methods available in standard LabVIEW as well as programming the controller for real-time processing by providing a real-time operating system.

The overall design of the software follows the State Machine architecture [6]. This is one of the fundamental architectures commonly used by LabVIEW programmers to implement complex decision-making algorithms that can be expressed as state diagrams or flowcharts. Each step of the program, e.g. Initialisation or buffering data, can be defined as a state of the system and depending on either user input or internal calculations will lead onto the next state. This allows for a very modular programming design where the processing is not fixed to a straight processing path but can quickly jump to different states when needed. This is achieved in LabVIEW by use of a case structure contained inside a while loop where each state of the program is a different case of the case structure as shown in Fig. 1. An enumerated value can then be used with a shift register to dictate which case or state will be processed next as a shift register will pass the data input to it to the next iteration of the while loop. Shift registers are also used to pass other data between the different cases of the case structure.

The cases defined for this system are ‘Ready’ where the data logger is waiting for the user to input the desired values for each of the operational parameters; sample rate, pre-trigger sample time (the length of time the user wants to be able to see a quench building up), post-trigger sample time (the length of time the user wants to see after the quench detection signal has been received) and the name to be used for the next log file. The GUI (called Front Panel in LabVIEW) is accessed using the NI Remote Panel Server [7] which allows the user to access the front panel through a web browser. As nearly all PC operating systems come with a web browser as standard this removes the need for each computer wishing to access the data logger having a LabVIEW program or executable that can connect to the variables and allow configuration.

Once the user has finished selecting these values and pressed the start button the program moves into the ‘Initialise’ case in which the rolling buffer is created based on the sample rate, the pre-trigger sample time and the post-trigger sample time. It is also the where the log file (of TDMS data type [8]) is created and the FPGA configured.

Next the data logger starts acquiring data and storing it to the buffer. The acquisition and some basic scaling of the data is done at the FPGA level as this provides the highly synchronised data acquisition and the high throughput needed to deal with 16 channels of data at up to 10kHz per channel. The output from the FPGA is stored in an internal FIFO buffer that can be accessed by the controller. Due to the limited space on this FPGA FIFO buffer the controller needs to read off the data in blocks of 300 samples and then store this in its own RAM buffer. This RAM buffer can provide for up to 450,000 samples per channel before needing to overwrite old data. This case in the program is also when the data logger is waiting to receive a trigger from the quench detection

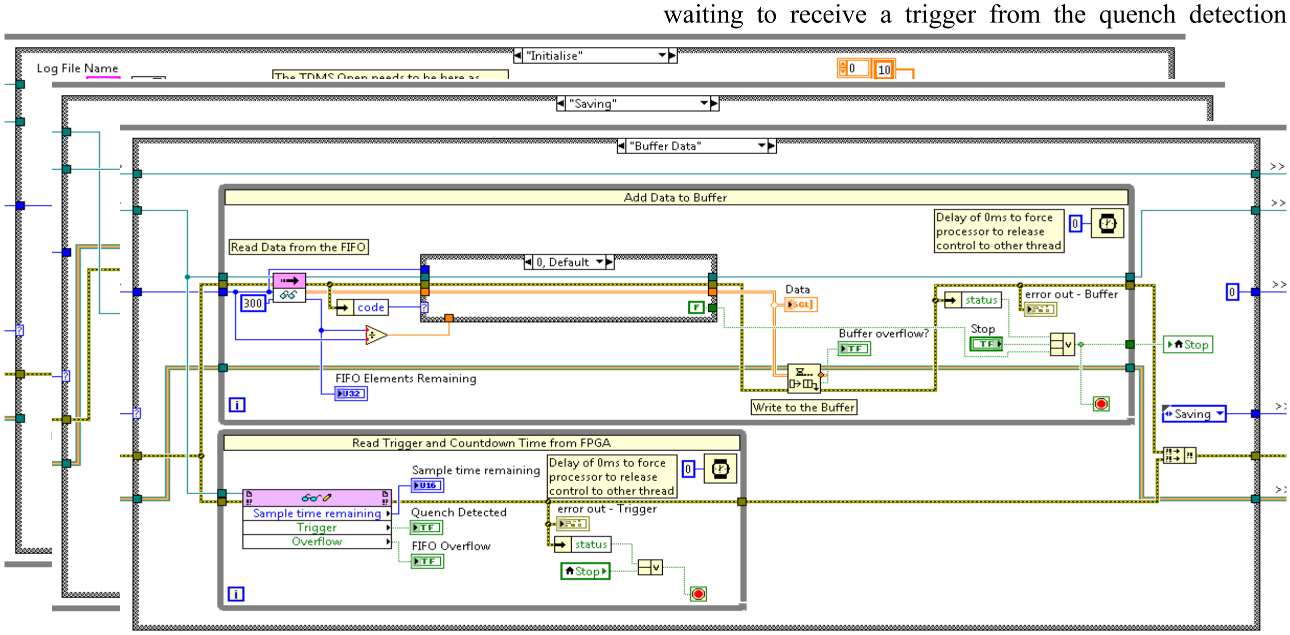

Figure 1: Three of the cases from the Data Logger case structure to illustrate the State Machine architecture.

ISBN 978-3-95450-146-5

system. This trigger comes into channel 16 of the data logger and is captured in the same way as the coil data as this allows for it to be saved in the log file synchronised to the coil data. To detect the trigger from the quench detection system a separate process runs in parallel to the data buffering, checking channel 16 for the voltage change that indicates a trigger. Once the trigger is detected the data buffering process continues for the post-trigger sample time specified by the user and then the program proceeds to the 'Saving' case.

The 'Saving' case simply takes data off the buffer and saves it to the TDMS log file created in the Initialise case.

Once all the data has been saved the program moves to the final case where it releases all of the resources it was using and deletes the DRAM buffer ready for the user to specify new values for the configurable parameters and start the logging process again. The users can then FTP into the non-volatile memory of the controller to access the log file.

RESULTS

The data logger was successfully used to prove the integrity of the quench detection system and to show that the magnets consistently, when pushed to the desired operational levels, suffer from a quench. When delivered the second focus coil magnet was tested under the same conditions, using the data logger as a diagnostic tool, and it did not suffer any unexpected issues proving that there was a fault in the first focus coil magnet. The first focus coil is now undergoing repairs to attempt to fix the issue.

FUTURE WORK

Due to the success of the data logger at helping to diagnose the fault in the focus coil it has been decided to integrate it into the final system as a permanent diagnostic tool. Other systems of the same design are being created to monitor other magnets on the machine as well.

Because of the limited size of the RAM on the NI 9012 controller care has to be taken to ensure that the desired buffer size does not exceed what the controller is capable of. Improvements are being made to limit the user to sample rates and sample times that the controller will be able to manage without error. Also, considerations are being made as to the added value of investing in a higher specification controller with more RAM available.

Usage of the data logger also pointed out another weakness in its design that is being addressed. As the data logger is designed to be a standalone system it is up to the user to extract the desired log files from it when they want them. However, this revealed a potential flaw as if the user was not interested in the log file from the last run, and thus did not extract it from the data logger and free up its non-volatile disk space then after a couple of runs the data logger has no more disk space to save new log files too. This causes it to enter an error state and lose the data. There was also a problem of the users starting the data logger running but then remembering to free up the disk space for the log file. When doing this after the data logger had already started the user could accidently delete the newly created log file from the controller causing it to enter an error state when it tried to save the data. Both of these issues are being addressed, firstly with a function in the Initialise case that checks to see how many log files are currently being stored on the data logger and deletes all but the newest ones, ensuring that there is enough space for new data to be saved. More thorough error handling is also being added in to recover from the error caused by the user accidently deleting the current log file by recreating a new file to dump the data to.

It is also desirable to have the data logger integrated with the existing EPICS control system [9] being used on MICE. Due to the limited RAM and processor time on the data logger's NI 9012 controller adding on the EPICS Server support software module that is available would have meant losing some vital buffering capacity. Because of this it was decided to make use of the on-board RS232 port. A simple serial protocol is being created using the LabVIEW VISA library [10] to allow EPICS IOCs to send values for the configurable parameters and start/stop signals. The data logger will in turn be able to send back information on what case it is currently running and any errors that have occurred.

CONCLUSION

This Data Logger system was successful in its purpose. The high specification analog acquisition hardware coupled with the FPGA chassis was able to give the high levels of synchronisation needed for this task. The limitations and difficulties of the system are being addressed within the scope of the future work to create a more robust tool that will be copied several times and is planned to be used throughout the MICE experiment for fault diagnosis.

REFERENCES

[1] MICE – Muon Ionization Cooling Experiment; http://www.stfc.ac.uk/208.aspx
[2] cRIO – Compact Reconfigurable Input Output; http://www.ni.com/compactrio/whatis/
[3] NI 9222 – 4 Channel Simultaneous analog inputs; http://sine.ni.com/nips/cds/view/p/lang/en/nid/209142
[4] NI 9103 – 4 slot 3M gate chassis; http://sine.ni.com/nips/cds/view/p/lang/en/nid/14158
[5] NI 9012 – Real-Time cRIO Controller; http://sine.ni.com/nips/cds/view/p/lang/en/nid/14158
[6] Application Design Patterns: State Machines; http://www.ni.com/white-paper/3024/en/
[7] Remote Panels in LabVIEW; http://www.ni.com/white-paper/4791/en/
[8] The NI TDMS File Type; http://www.ni.com/white-paper/3727/en/
[9] EPICS – Experimental Physics and Industrial Control System; http://www.aps.anl.gov/epics
[10] NI VISA – National Instruments VISA; http://www.ni.com/visa/

ISBN 978-3-95450-146-5