Design Note for the Single Circuit Quench Protection Monitor (uQPM)

Kevin S. Martin, John Diamond, AD EES, Fermilab, Batavia, IL, 60510, USA

Introduction

The Single Circuit Quench Protection Monitor (uQPM) has been designed to provide quench protection for a single super-conducting circuit with from one to four cells. Included is absolute and relative quench protection, over-current protection, and lead voltage protection. This is done for a cost of <$1000 not including embedded costs (i.e. chassis, wiring, DC power supply).

Hardware

The hardware that has been chosen is the Prometheus from Diamond Systems. It is a PC/104 single board computer with a 100Mhz x86 CPU, 32MB RAM. The Prometheus has data acquisition hardware on-board so no additional DAQ boards are needed. It has 16 multiplexed 16-bit ADC channels with a 100KHz converter, 4 12-bit DAC channels, 24 DIO, 2 counter/timers.

Software

Real-Time Operating System

The RTOS that was chosen for this project is eCos. eCos is an open source, royalty-free, real-time operating system intended for embedded applications. vxWorks was ruled out because of costs (architecture support, BSP, and target license fees). We initially attempted to implement this application using real-time LINUX (Timesys LINUX). However, it proved, at least with the hardware we were using, not to be responsive enough (worst-case interrupt latencies were >3mSec).

The development host for eCos is a LINUX box called bdeelinux.fnal.gov. The eCos Tools come with a GUI application as well as a command line utility which are used to configure and build eCos kernels. Once you have built a kernel (for the specific CPU and board you are using) you link your application to the kernel. This then creates a monolithic kernel-application object. The object is then placed on the target's flash disk for booting by GRUB (target's boot loader). Alternatively, Redboot can be used to load the object over the network when network debugging (GDB) is desired.

Error! Not a valid link. Control System Connectivity

One of the drawbacks of choosing eCos as our RTOS and not vxWorks is that we had no easy way of connecting these embedded systems to the ACNET Control System. The AD Controls Department ACNET/MOOC software only runs under vxWorks.

The solution to this problem was to implement an intermediary vxWorks/MOOC/ACNET system which would act as a communications bridge to/from the control system for these low cost embedded systems: BADMAB. We have implemented an Instrumentation Front End Protocol called BackDoor in the eCos based embedded systems. This protocol was developed by Duane Voy and is a light weight, object based protocol written in C++. See the documentation for the BackDoor protocol at: TheBackDoorSystem.pdf

The bridge system (BADMAB) passes on ACNET requests to the eCos based embedded systems using this BackDoor protocol (see Figure 1). Many BackDoor Server systems can be supported by a single MOOC/BackDoor Bridge System. ACNET Readings, Settings, Digital Status and Control, and Fast Time Plotting are currently supported using this mechanism. A special ACNET SSDN was designed to allow ACNET devices to specify which BackDoor Server systems to reference (See Figure 2). The actual Backdoor-MOOC bridge system is called badmab.fnal.gov. It is located in the Transfer Gallery Micropit (TGS-120).

Design Note for the Single Circuit Quench Protection Monitor (uQPM)

Kevin S. Martin, John Diamond, AD EES, Fermilab, Batavia, IL, 60510, USA

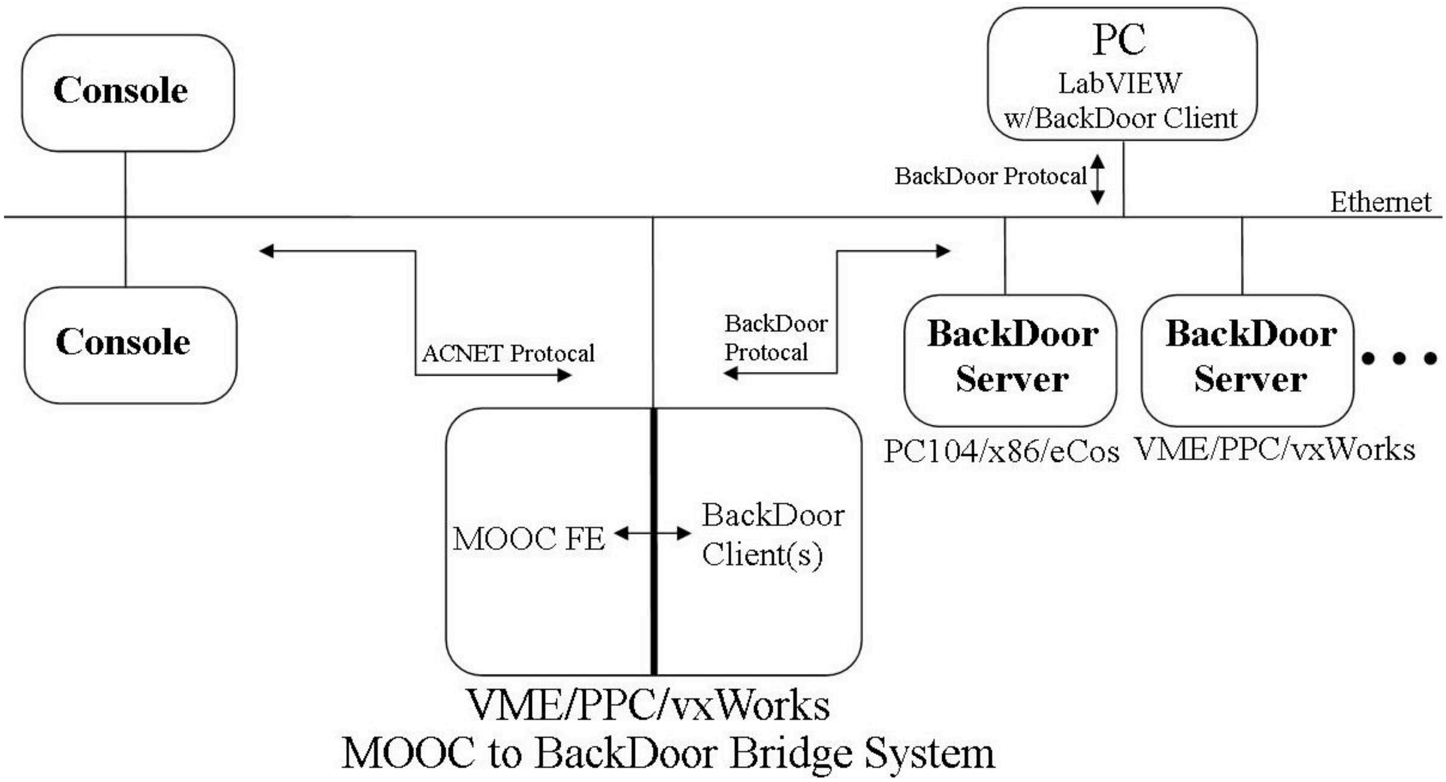

Figure 1
BADMAB System Architecture Diagram

ACNET ID		MOOC OBJECT ID		P1	P2		CLASS	CHANNEL
0	1	2	3	4	5		6	7
ACNET ID		MOOC Object ID / BADMAB Client ID		Misc.	F T P	FTP Accessor ID	Accessor ID	Channel Index

ACNET ID	Unique for each BADMAB ACNET Node
MOOC Object ID	Unique for each BADMAB BackDoor Client
P1	TBD
Fast Time Plot Enable Flag and Accessor ID	BackDoor Fast Time Plot Accessor number. The most significant bit of this field is the Fast Time Plot (FTP) enable bit.
Accessor ID	BackDoor Accessor number.
Channel Index	Index into data read/written by an BackDoor Client Accessor.

Figure 2
BADMAB SSDN

Design Note for the Single Circuit Quench Protection Monitor (uQPM)

Kevin S. Martin, John Diamond, AD EES, Fermilab, Batavia, IL, 60510, USA

Source Code Repository

The source code is all located in the Accelerator Division CVS code repository on nova.fnal.gov. It is organized as several different CVS projects:

rfiinst/shared/backdoorserver - vxWorks version of the BackDoor Server
ees/backdoorserver/ecos - eCos version of the BackDoor Server
rfiinst/shared/backdoorsupport - vxWorks version of the BackDoor portability layer
ees/backdoorsupport/ecos - eCos version of the BackDoor portability layer
rfiinst/shared/backdoorclient - vxWorks version of the BackDoor Client
ees/promdaq - Hardware data acquisition driver library for the Prometheus board
ees/ecosutils - Library of the common eCos utilities and BackDoor Accessors
ees/badmab - Source code for the BADMAB project
ees/uqpm - Source code for the uQPM project

Single Circuit Quench Protection Application Software (uQPM)

The uQPM quench protection algorithm runs at 1000Hz. The Cell Voltages, Current, Lead Voltages are sampled once[1] every mSec by the ADC. The ADC hardware then causes an interrupt which causes an Interrupt Service Routine (ISR) to be called. This ISR scales/filters the input signals, does the various computations, stuffs the circular buffer, and then also makes the decisions on whether there have been any trips or reset requests etc.

A diagram showing the complete uQPM algorithm is at the bottom of this document. The uQPM assumes that the equivalent circuit of the load is two inductors in series with a resistor (R_2) across one of the inductors (see Figure 3). This assumption allows the uQPM to better match the load current and voltage during transient conditions and thus do a better job at quench protection without unnecessary "false" trips.

Therefore, the total magnet inductance (for DC current) would be roughly equal to the sum of L_1 and L_2. Roughly because these parameters have been tuned while applying sudden voltage transients to the load and adjusting R_2, L_1, and L_2 to give the smallest quench signal possible. Thus, there is some trade off made in optimizing both the "fast" and "slow" responses. Some quench circuits have a series resistance so R_1 is used to account for this. $V_{cell} = V_{measured} - (R_1 - I)$ where $V_{measured} = V_{cell1} + V_{cell2}$. If a particular quench circuit has no series resistance then R_1 is parametrically set to zero.

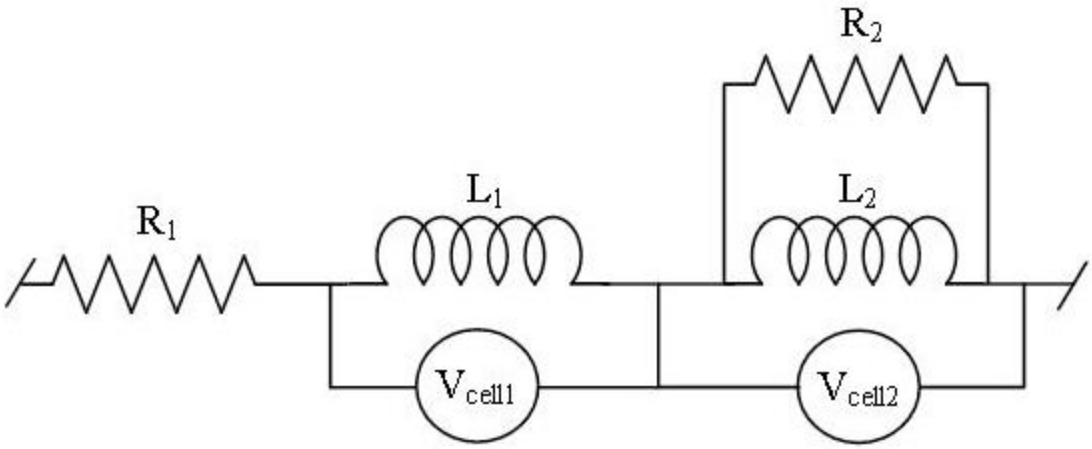

Figure 3

[1] Actually the Cell Voltages and Current signals are over sampled six times. These six over sampled values are then averaged. This over sampling is done to reduce the amount of ADC digitization noise. We have found that the noise can be reduced from 2-3 bits to 1-2 bits by using this method, the more over sampling done the better (up to a point). In the uQPM, six times over sampling was chosen as a balance between results and time/processing overhead.

All the input signals as well as the results of the various computations are stored, each cycle, in a two second deep, circular buffer. This circular buffer is used for two purposes. First, when ever a trip occurs a Trip Buffer is saved. These Trip Buffers are the complete circular buffer for 1 second before and after the time of the trip. The uQPM will keep up to 10 of these Trip Buffers along with the time-of-day when they occurred. An application (not a Console Application) is available for reading/plotting these Trip Buffers. Second, the circular buffer is used for doing real-time plotting both with a LabVIEW application as well as through ACNET using Fast Time Plotting (FTP). FTP is supported at up to 1000Hz.

A spreadsheet listing all the ACNET devices for each of the TEL2 uQPMs, with descriptions, is shown in Table 2. A spreadsheet listing all the default values of the ACNET devices for each of the TEL2 uQPMs is shown in Table 1.