

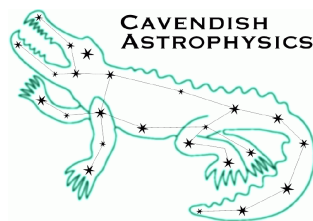
MRO Delay Line

VME Software Functional Description

R.C.Boysen
roger@mrao.cam.ac.uk

rev 0.1

06 July 2007



Cavendish Laboratory
JJ Thomson Avenue
Cambridge CB3 0HE
UK

Objective

To describe the overall architecture of the VME software.

1 Introduction

The VME sub system compares the position of the cat's eye with the demanded position as received from the workstation and outputs a control voltage to the cat's eye via the low latency link. It also sends commands to the trolley micro to switch between track mode and slew mode.

2 Metrology hardware

The metrology system is based on the Zygo displacement measuring system (ZMI 2000) and the readers is referred to the web site (www.zygo.com) for more technical details and the principle of operation.

2.1 The hardware consists of

1. Laser head (7702)
2. Linear interferometer (7002)
3. Retroreflector(7003)
4. Various mirrors
5. Beam expanders
6. Fibre optic pickup
6. Measurement board(2002)
7. Datum switch(Sony SET-P15-1)

A schematic layout is shown below in Fig 1

Optical fibres connect the receiver to the measurement board and from the laser to the reference input on the measurement board.

The datum switch is a magnetic sensor operated by a magnet on the trolley. This is connected to a electronic latch which can be reset.

The VME chassis contains:-

- 1 Processor (Concurrent Technologies CPU - x86 with Ethernet port, keyboard, mouse, VGA port, etc),
- 2 Timing board,(Symmetricom TTM635VME)
- 3 Measurement board,(Zygo 2002)
- 4 Industry pack carrier board with DAC and PIO,
- 5 Interface board.

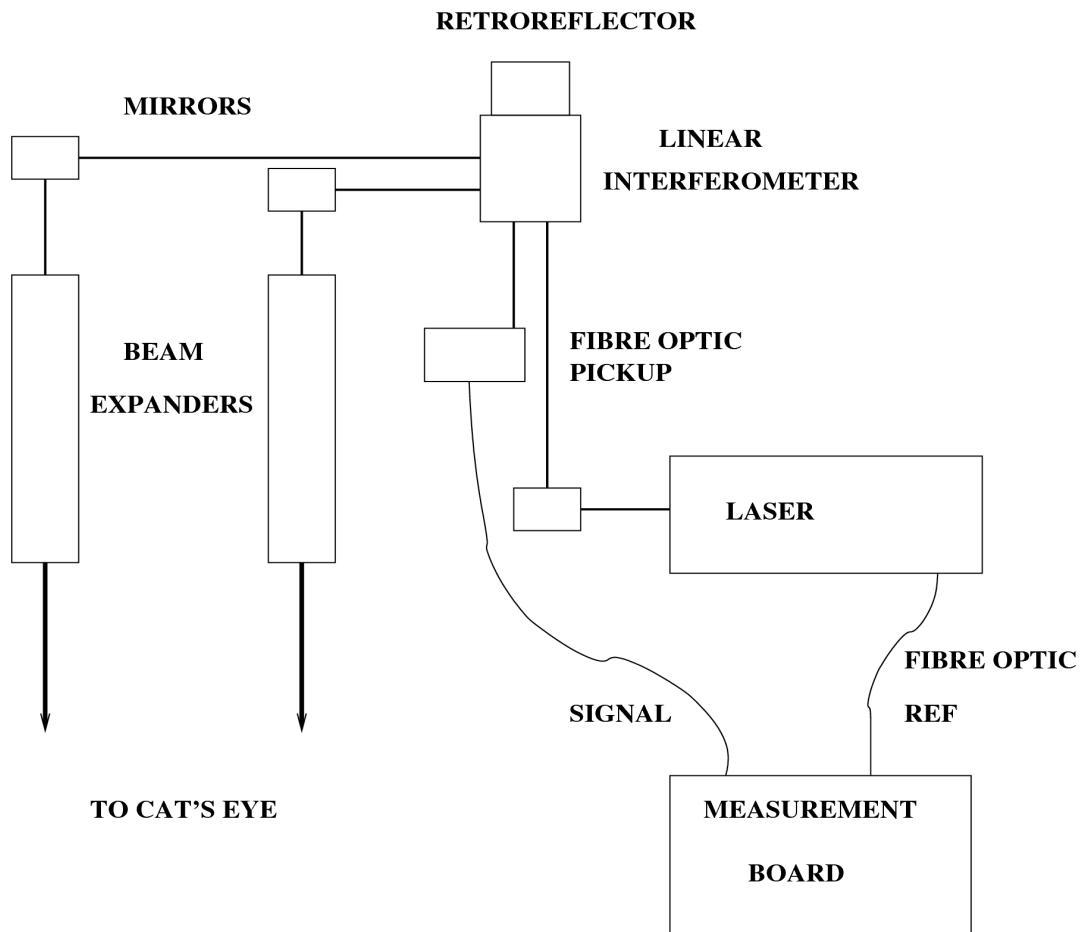


FIG 1

3 Development and Execution Environment

The operating system is QNX 6.1 This was chosen because it is a real time operating system with a fast, deterministic interrupt response time and also our previous experience at COAST. The interrupt response is essential in optical interferometers as the cat's eye must be positioned and track to sub-micron accuracy. At maximum tracking speed of 15mm/sec a timing error of 10usec translates to an error of 0.15 microns in cat's eye position. QNX is UNIX-like and follows various industry standards such POSIX and TCP/IP. The system is used to both develop and run the software which is written in C.

4 Program Architecture

The software is divided between several threads or processes which run concurrently. One thread deals with messages to the workstation; another deals with interrupts and there are two threads to deal with command and data messages from the workstation. The advantage of using threads is that if there is no interrupt/message than the other threads are free to run but immediately an interrupt occurs or message arrives that thread will run.

5 Main program

This routine sets up a common data area for use in the interrupt routine, initializes VME hardware, initializes status and telemetry messages and establishes connections with the trolley in order to send commands. It then starts two threads to deal with command and data messages and one thread with a higher priority to deal with interrupts. Note that eventually there would also be another thread to deal with messages from the fringe tracker. The program then loops sending status messages to the workstation and commands to the trolley every tenth second and telemetry data to the workstation every second.

6 Interrupt Routine

The interrupt routine comprises two parts; the interrupt service routine (ISR) which runs when a hardware interrupt occurs and a second part when the ISR returns.

The ISR uses complete resources of the processor and therefore runs deterministically so that a minimum time elapses between sampling the measurement board and outputting a drive voltage to the cat's eye.

The timing board generates a signal at 5KHz rate which samples the Zygo measurement board and is connected to the 'SYSFAIL' signal on the VME bus. This method of generating the CPU interrupt was chosen, rather than a VME bus interrupt, because the 'SYSFAIL' signal is fed directly to the processor interrupt lines and so gives minimum response times.

The ISR immediately reads the measurement board, compares the value with the previous value to check if the measurement board has overflowed. (See note 1) The routine then compares the position with calculated destination, adding the fringe tracking offset if that is enabled. If the error is outside fixed limits, the system switches to slew mode, otherwise the velocity for the next 5 KHz interval is added and the value is transferred to the DAC. Finally the position and velocity for the next interrupt are calculated.

If the interrupt is at the 'second' (of time) the interrupt count is reset and a 'second' flag is set.

When the ISR return the telemetry data is stored in a buffer and every tenth of a second, new position and velocities are read from the received data buffer for action by the next ISR.

If the 'second' flag is set, the 'telemetry buffer full' flag is set indicating that this block of data should be sent to the workstation and the storing of telemetry data is switch to the second buffer.

If a command to go to the datum has been received the measured position of the cat's eye is ignored and the trolley switched to slew and driven to the datum position.

Note 1

The Zygo measurement board is designed to count +/- 21.2 M and has 36bit storage so for 200 M an extra 4 bits are required which are added in software. The board has to be programmed to carry on counting after an overflow has occurred.

7 Datum routine

The datum routine is entered on a command from the workstation. This is necessary at start up and if the metrology loses count for any reason.

The sequence is as follows:-

1. Reset the datum latch circuit.
2. Slew at maximum speed towards datum
3. Stop the trolley when the datum latches
4. At this point the trolley will be past the datum switch
5. Reset the latch and drive slowly back until the datum latches
6. Continue driving the trolley until clear of the datum switch
7. Drive very slowly towards datum until switch operates
8. Reset metrology counter and re-close cat's eye servo loop

If the trolley is already past the datum when commanded to go to the datum then the trolley will drive into the pre-limit. The workstation will then drive the trolley out beyond the datum by a direct slew command to the trolley micro and the above sequence started.

8 Message Routine

The format for sending and receiving messages has been described fully in the document 'Control Software Architecture'

Command messages are checked for which trolley they apply to and the appropriate flag set.

In the case of data messages, each message, containing the data together with a time stamp, is stored in a circular buffer that can store up to ten messages. Messages are sent at the rate of 1Hz and contain ten positions and velocities. When a one second interrupt occurs, the time stamp of the messages in the buffer are checked and the appropriate set of positions and velocities used in that second. Hence data can be sent sometime in advance and will not be affected by network delays.