ERROR HANDLING IN THE NSLS CONTROL SYSTEM

Susila Ramamoorthy, Pauline Pearson and John Smith
National Synchrotron Light Source, Brookhaven National Laboratory, Upton, NY 11973

Abstract

The error handling software is an important component of the NSLS control system and has been in use since 1982. Following the major control system upgrade, the error processing software has been improved at both micro and workstation levels. This note describes strategies used in error detection and reporting and the workstation software used for display and analysis of error messages.

I. INTRODUCTION

The control system at the National Synchrotron Light Source Facility has been upgraded to meet the increasing demands on the data acquisition speed and CPU power. The upgraded system has a two-level distributed architecture consisting of HP workstations for the host-level and VME-based microprocessor systems for the equipment level. The communication link is ethernet[1,2]. The VME micros are responsible for the hardware control and data acquisition. The Error handling system, initially installed in 1982, has been modified to work with the upgraded system. A lot of effort has been put into the error detection and error reporting software to make the Error Handler System a useful diagnostic tool for the operation of the facility.

II. OVERVIEW

The micros continuously monitor the associated hardware for error/alarm conditions and send error messages to an Error Processor which is itself a real-time micro system. The Error Processor displays the messages as they are received, on a console in the control room. In addition, the error messages are stored in memory. An Error Logger running as a background process on the main workstation periodically retrieves the error messages from the Error Processor micro and saves them on a disk. Figures 1 and 2 represent the physical layout of the Error Processor system and the data flow respectively. In the following sections, the micro software for error reporting, the Error Processor micro and the workstation software packages for error analysis are described.

III. MICRO SOFTWARE

All the VME micros are driven by a real-time operating system referred to as NSLS Control Monitor [3]. The monitor consists of a set of system tasks and interrupt handlers, built around a commercial real-time kernel (RTUX by Emerge Systems Inc.) and application/device-specific modules. The system software which is standard for all micros, manages communications, command decoding, displays, etc. It provides a utility library and services for error detection and reporting. The device-specific tasks are unique to each micro and are dictated by the hardware associated with it. The monitor views the hardware as a set of logical devices with unique names. A logical device is a software data object defining the data structure and the commands for a hardware or software parameter controlled by a micro. It responds to a standard set of commands from the workstations. One logical device can be mapped to one physical device or one logical device can represent multiple physical devices or vice versa. A standard format is used for all logical devices.

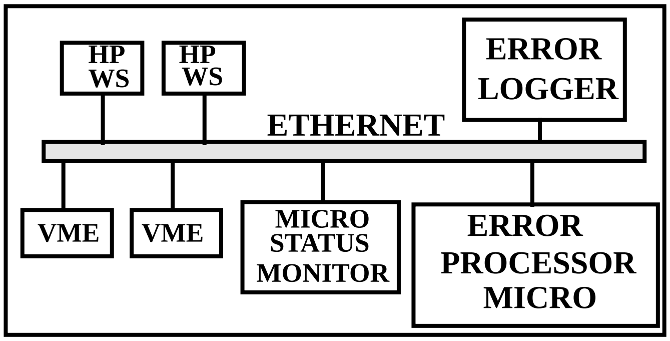


Figure 1. Physical Layout of the Error Handler System.

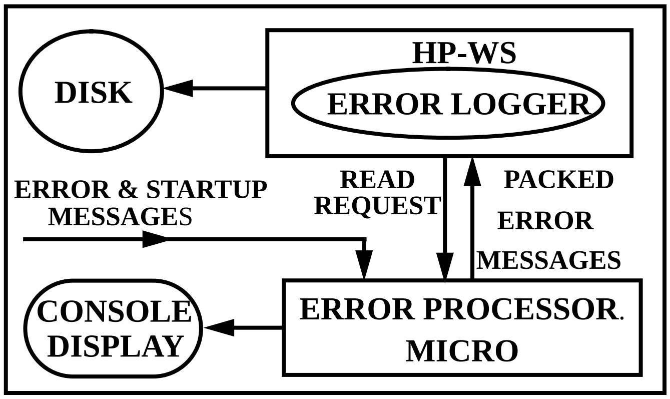


Figure 2. Data flow in the Error Handler System.

The relevance of the logical device concept in the current discussion is that the device structure contains all information necessary for error detection. The device-specific tasks scan the device hardware at 10 Hz or higher for alarm conditions and report errors using the services provided by the system part of the monitor.

The types of errors depend on the function of the logical

*Work performed under the auspices of the U.S. Dept. of Energy under contract no. DE-AC02-76CH00016.

device. Following is the list of alarm conditions the software looks for:

1. Tolerance error for analog input-output devices when the difference between the set-point and read-back is not within the specified tolerance range. (e.g. The difference between the desired current for a power supply and the actual current exceeds the tolerance limit).

2. Alarm conditions when the monitored data falls out of the safe limits. (e.g. The data monitored by a radiation monitor is out of the specified range).

3. Time-out conditions observed while controlling an instrument or while acquiring the data from an instrument. (e.g. No reply is received when an instrument is read/controlled via an RS-232 serial port).

4. Invalid state error when the state of the digital device is different from the commanded state. (e.g. A shutter is commanded to open but status indicates closed).

5. Fault state of a digital input bit device. (e.g. Water flow error, RF cavity fault etc.).

The following features of the micro software help to identify true error or alarm conditions and prevent notification of meaningless and redundant messages:

1. Proper strategy is exercised for error checking and reporting. For all devices when a set command is received, alarm checking is inhibited until the command is executed and the hardware it represents settles. The settling time may vary from millisec. to seconds. The device is marked In-Process until the settling time has elapsed.

2. Alarm/fault states are not recognized unless valid conditions are satisfied or the hardware is in a particular state. As an example, a number of power supplies can be controlled by a power controller. In this case, tolerance error check is invalid unless the controller is on. The software checks this condition before monitoring the power supplies. Another example is the various faults associated with an RF amplifier. The “filament_under_current” or “plate_under_voltage” conditions are not real alarms unless the RF amplifier has been commanded “ON”. The software uses the appropriate logic before checking for errors.

3. After detecting a fault, an error message is sent asynchronously to the Error processor. The message contains all information about the device such as the micro name, logical device record number, message code, time stamp and all pertinent data. The types of error (tolerance, time-out, etc.) are stored in the auxiliary status bytes of the record.

4. The error flag is latched in the device record to prevent error storms. No errors will be reported for the device until the error status is cleared.

5. Errors can be reset in a number of ways. For devices that accept “SET” commands, the monitor resets the errors when a SET command is received. The operators at a workstation can issue a clear command to any device or a global clear error command to a micro. When the error flag is cleared, if the device is still in error, an error message is generated again. If the device is out of alarm state, an Error Reset message is sent to the error processor micro. The monitor keeps updating the count of devices that are in error.

6. Disabling of error reporting can be done by simple commands from the host level computers. The error detection and error latching are still effective and the error status of the device can be obtained by reading the device record from a host level computer. This feature is useful when a particular hardware is known to be malfunctioning and the operator wants to shut off the alarms for those devices associated with the hardware. Error reporting can be re-enabled from the workstations. Error reporting can be disabled on a micro basis by selecting the configuration switch in the micro. This is required during the maintenance period when hardware is being checked out.

Alarm parameters such as tolerance value, limits for alarm level 1 and alarm level 2 can be modified by authorized personnel from any host level computer. Similarly the calibration values for read-backs can be changed to accommodate aging or changes in hardware. These parameters are saved in non-volatile memory and can be retrieved even if the micro is re-booted. The software also provides the option to restore the original or default values on re-boot.

In addition to the alarm/error messages, the micro sends a start up message when it is reset or powered down. The micro determines whether it is a COLD start or a WARM start and sends the appropriate message to the error handler. A cold start indicates that a micro has been powered down; a warm start indicates that a micro has been reset.

IV. ERROR PROCESSOR MICRO

This is similar to any other real-time micro in the system. All micros send error or informative messages using the micro to micro communication utility software in the monitor. The messages have protocols identical to those from workstations. The messages are similar to the “Set Data Array” packages from the workstations. The Error Processor decodes the message and generates a scrolling display on a console in the control room. Each message is displayed as a line and contains the micro name, the device name, description of the error or information with time stamp. Every 2 minutes, a blank line is written to scroll old messages off the display to ensure that the new messages can capture the operators attention. In addition, a beep is generated for each message to alert the operators. The messages are stored in a ring buffer in chronological order. These messages are retrieved by the Error Logger process on a workstation.

Another key system is a Micro Status Monitoring micro. This periodically polls all the micros and generates a TV display of all micro names in green. If a micro fails to reply, the name is highlighted in red. The TV display has been extremely useful since one can immediately notice the failure of any micro and fix it. This reduces the loss of History data collected by the History programs on the workstation. Also, the micro can be fixed before the next beam injection into the storage ring thereby reducing the machine down time. Following a power dip, all micros automatically re-boot themselves. Any micro that does not re-boot can be identified immediately and reset manually.

V. ERROR LOGGER

This process runs in background on a workstation. Every 5 seconds, it collects the error message packets from the Error Processor micro and stores them in binary files on a disk. A new file whose name extension has the encoded date-time stamp is created at 00:00 hours every day and the messages for that day are stored in the file. If the error logger is restarted for any reason and a file for that day does not exist, a new file is created.

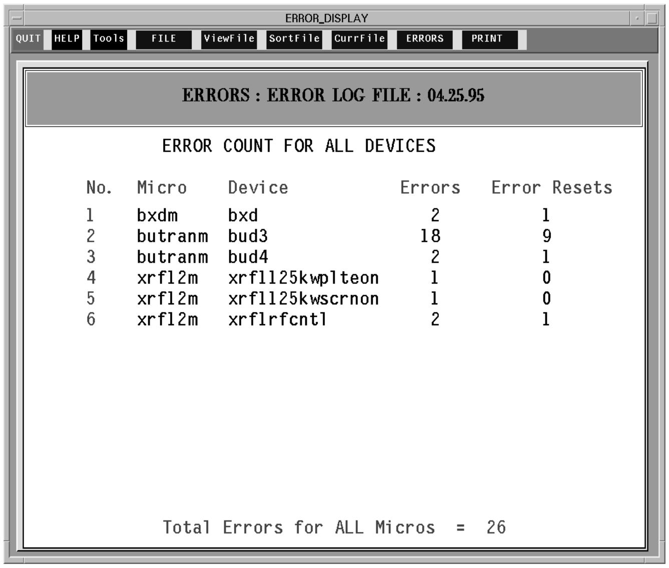
VI. WORKSTATION SOFTWARE

All workstation programs are X window Motif based. A set of tools is available for analysis and display of error messages and to monitor the network message rate.

Using standard data acquisition programs [4], one can get all parameters of a logical device. If a device is in error, the error field will be highlighted.

One of the packages referred to as Microstatus retrieves the Current Status array from all or a specified micro and displays the total number of devices in error, spurious bus interrupts, version number, the time elapsed since last re-boot, the list of devices in error etc, at any time.

Another important package is the error display and analysis program. This program can be run on any workstation. The program displays a menu bar with various options (Figure 3). By selecting the “FILE” menu one can get the list of the error log files and select the one for the required day. Old error files can be deleted by authorized members using this program. One can view the entire day file or a given time slot. The messages will be displayed with the micro name, device name, timestamp, type of message and all the relevant data at the time the alarm was captured. There is also an option to view the current messages as they are logged on to disk. Other options allow one to get different types of summary: 1. Sort devices that are in error in the alphabetical order of micro names, 2. Summary of total number of error for all micros, 3. Number of errors for each device in a micro. One can get a hard copy of all the displays using the print menu button.



```
ERRORS : ERROR LOG FILE : 04.25.95

          ERROR COUNT FOR ALL DEVICES

     No.  Micro    Device           Errors   Error Resets
     1    bxdm     bxd                 2          1
     2    butranm  bud3               18          9
     3    butranm  bud4                2          1
     4    xrfl2m   xrfl125kwplteon     1          0
     5    xrfl2m   xrfl125kwscrnon     1          0
     6    xrfl2m   xrflrfcntl          2          1

          Total Errors for ALL Micros  =  26
```

Figure 3. Error Analysis and Display

A network test program allows one to send messages to any micro as fast as it can respond and prints out the communication rate, count of lost or timed-out messages, retries etc. This data helps one to identify the hardware problem (in the network or in the micro) or software problems (some unpredictable delay in an interrupt handler etc.).

VII. CONCLUSION

With the exercise of proper strategies in error detection and reporting in the micro software, meaningless errors have been eliminated and the error handling system has become a useful diagnostic tool both from the operations and engineering point of view. A number of hardware failures have been identified and fixed. Following a problem, one can often reconstruct the events that led to the failure. The micro status monitoring micro with a TV display has been extremely useful in minimizing the down time and loss of history data. The future plan is to incorporate the micro status monitoring function in the error micro itself. In future, the workstation programs would be able to send messages to the Error Processor to notify error or warning messages to the operators and to log them into the error message file.

VIII. REFERENCES

[1] J.Smith, et al. Proc. IEEE Part. Acc. Conf. (1993), 1852.
[2] J.Smith, S.Ramamoorthy and Y.Tang, Nucl. Instr. and Meth. in Phys. Res. A352((1994) 114-117.
[3] S.Ramamoorthy and J. Smith, Proc. IEEE Part. Acc. Conf. (1993),1849.
[4] Y.N.Tang et al., Proc. IEEE Part. Acc. Conf. (1993),1846.