PRINCIPLES OF STATUS CONTROL AND INTERLOCK PROTECTION SYSTEM FOR THE SRS

S. V. Davis, M. T. Heron, A. Oates, B. G. Martlew, A. Quigley
CCLRC Daresbury Laboratory, Warrington, UK WA4 4AD

Abstract

The status control and interlock protection system of the Synchrotron Radiation Source (SRS) uses close coupling of the plant equipment and its associated interlocks. It exists in three forms but the operating principles are common across systems. Functions ensure that a piece of plant cannot be operated if all the interlocks guaranteeing safe operation are not present. A handshake protocol is then used to ensure that the equipment is correctly started. In addition to interlock protection, all systems deploy at least one form of watchdog protection should hardware or software failure occur plant would be safely shutdown. The status control model is used for most controlled equipment giving a familiar operator feel to the equipment control process. This in-house design philosophy has proved cost effective and reliable and could be continued for future accelerator control projects.

INTRODUCTION

The SRS is a second–generation 2 Gev synchrotron radiation source that has been operational and continually developed over the last 20 years. Traditionally, control of vacuum pumps and instrumentation, together with isolation valves, radiation masks and station shutters has been integrated into the main SRS control system[1]. This system was originally designed to interface to simple, non-intelligent devices requiring nothing more than ON/OFF/RESET control operations together with 16 hardware interlocks and the option of analogue control and monitoring via DAC's and ADC's.

Modern intelligent equipment with serial bus control is now widely used, but it has been judged that it is unsafe to rely on a serial bus for plant protection. Therefore the current status control and interlock protection system is still relevant today and proposals exist for integration into the EPICS control system. [2]

The original status system used TTL logic driven state machines designed into CAMAC crates while the second generation then moved to embedded G64 processors [3]. The current system uses VME OS9 Front end computers (FEC) and CAN bus Status modules. All three generations of status system employ similar principles of operation, but the two earlier systems are of historical interest only, therefore this paper will concentrate on the latest system.

THE FRONT END COMPUTER (FEC)

The status control stations use a 3U VME system running OS9. The VME system runs a number of processes, three of which are specific to the control system applications[3]: a server process to the higher levels of the control system, a process to communicate with the CAN Bus and implement the status and interlock functionality and a process to communicate with intelligent instruments over serial communications, see Figure. 1.

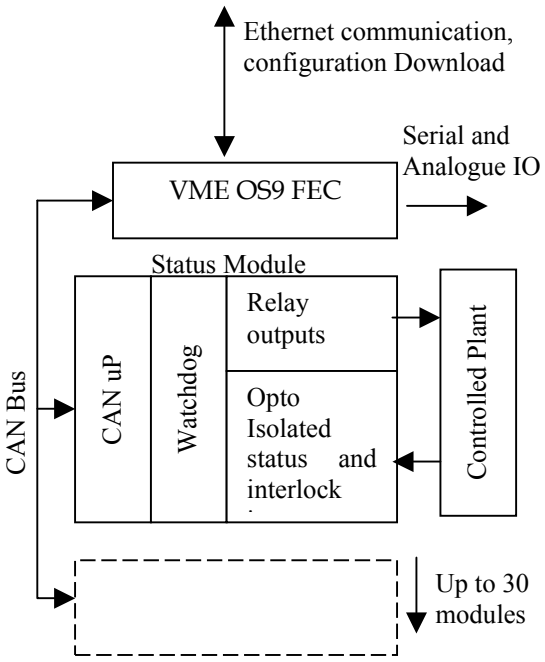

Figure 1: CAN Bus Status hardware

Status Database

Executable files are downloaded from a server to each FEC on boot up. One file creates an OS9 memory module known as the "status database". A configuration file is then executed which inserts the correct entries for all the status equipment controlled by that FEC.

All the data from the hardware passes through the database. Each control module has a series of entries in the database, which select a software driver, mask interlocks, set up timer values, and patch interlocks.

Patched Interlocks

The ability exists to software patch interlocks between control modules. This allows a plant ON to become an interlock in another module. For example the ON signal from a downstream valve patched to the preceding radiation stop would hold the stop closed until the valve has fully opened. Interlocks can be directly patched to other modules to minimise wiring. Therefore a vacuum pressure good interlock can be wired to one module and

be software patched to many other modules. The interlocks are patched at the status driver level. A series of entries in the database specify if interlocks need to be patched into other modules. The source and destination pointers are listed in the database.

Status Driver

A 'C' function defines how the status module interacts with the equipment type being controlled. It communicates only with the status database, leaving hardware changes in a database location for the main supervisory programme to transfer to output. Currently only four driver types are used - General Purpose, Valve, Titanium Sublimation Pump, and Auxiliary.

Status Tasks

An OS9 memory module is used as a configuration database and for storing interlocks and module status. The status software performs the following tasks – (1) Each status module is read in turn and the database updated with interlocks and plant reply. (2) Process interlocks to other modules (3) The device driver that defines the desired behaviour is executed, the driver name being specified in the database. (4) Finally the control output is read from the database and written to the status module. (5)This process continually repeats, cycling through all the installed status modules. This process is illustrated in figure 2.

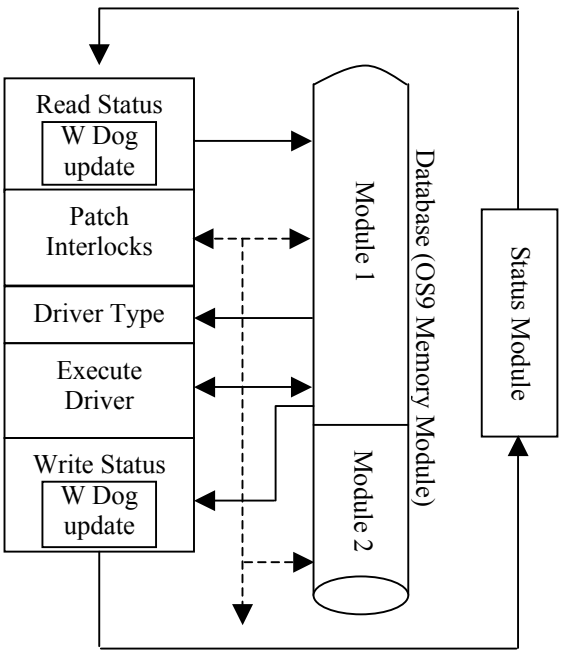

Figure 2: Software Execution Model

GENERAL STATUS MODEL

To allow plant operation all the conditions shown in Figure 3 need to be met. The status message reports what, if anything, is preventing the switch on. The messages shown are for the latest CAN Bus controlled status interface. Similar hardware messages exist in the previous generations of hardware.

CAN BUS ERROR – There is no communication to the CAN bus status module. The plant must be off because the status module requires a watchdog update.

W DOG FAIL – Each status module has a built-in watchdog timer which must be updated every 160mS by toggling a bit written over the CAN bus. This condition is latched and appears after a power cycle.

UNPLUGGED – Reports a continuity link, which can be wired throughout the cabling to the controlled equipment. It indicates if any part of the control cabling is unplugged.

OFF FAULT – All the previous conditions met but one or more interlocks are bad.

OFF READY – No fault conditions exist; the plant is ready to be operated

TIMING - An ON command has been received and the plant has been requested to operate but the plant has not replied to indicate that it is in stable operation.

ON – ON REPLY has been received from the plant to indicate stable operation.

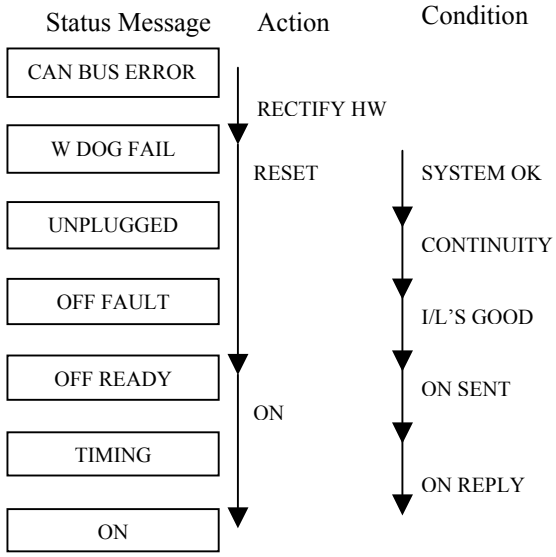

Figure 3: Status Diagram for a general-purpose status interface

Each of the status messages can be mapped to a device specific message in the higher levels of the control system. For a valve OFF FAULT – SHUT FAULT,OFF READY – SHUT READY, TIMING – OPENING and ON – OPEN. In the case of valve control *on reply* would be connected to the open limit therefore if the valve fails to reach its open limit then the status display would show OPENING then return to SHUT READY.

Switch On Logic

Figure 4 demonstrates the logic principles, which allow the plant to be operated and shut down.

The ON REPLY operates like a handshake from the plant holding the equipment on. If ON REPLY is lost then the status returns to OFF READY.

The timer is variable to allow for the response time of the controlled equipment. In the case of a valve it can be many seconds but a PSU could respond in less than a second. The timer value is a database entry, which is adjustable in 40mS steps.

In the event of plant being forced into the ON state by other than the control system the control system will move into the ON state when it receives the ON REPLY, but if any interlocks are bad it will report INVALID.

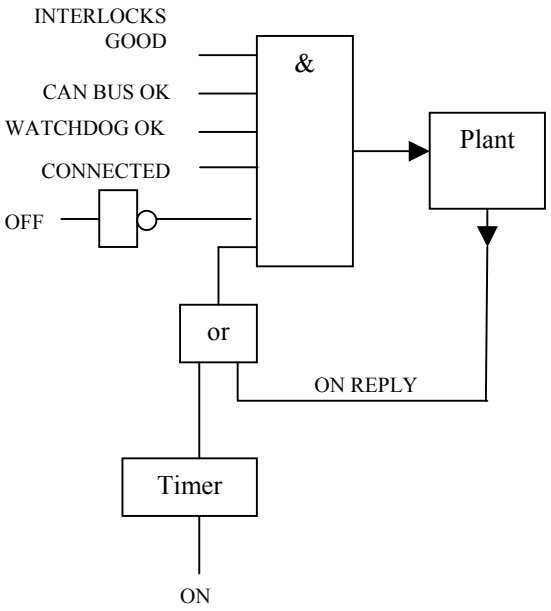

Figure 4: Plant control logic diagram

In early systems these functions were implemented in TTL logic, but in current systems C code defines the behaviour.

INTELLIGENT PLANT CONTROL

Modern microprocessor controlled equipment is generally controlled over a serial bus. The FEC configures and controls serial equipment in a similar manner to the status control making use of a database and supervisory process and drivers for the equipment type. When plant protection needs to be assured the status system is used to enable the equipment via an external interlock or in the case where there is no hardware interlock then by controlling the main power tothe plant.

This results in two control parameters for one instance of the equipment. The extra parameter is called the auxiliary and has only one command, RESET, which will reset any latched bad interlocks and give a closed contact to enable equipment operation. When the auxiliary reports good the equipment can then be operated with the second parameter which controls via the serial port. When graphical display panels are designed the auxiliary is placed next to the control parameter.

CONCLUSION

The status control model is used for most controlled equipment on the SRS giving a familiar operator feel to the equipment control process. This in-house design philosophy has proved cost effective and reliable and could be continued for future accelerator control projects.

FUTURE DEVELOPMENT

The CAN Bus status modules and the CAN protocol have been proved to be highly reliable and easy to use. It is intended to control them via an EPICS IOC under VxWorks.

This development is intended for use on 4GLS [4] a Fourth generation light source, work has already started a Daresbury on a pilot machine for developing the principles needed to construct the main machine.

One area that will need to be investigated are whether the EPICS scan function will be reliable enough to maintain watchdog operation or will another process or interface layer will be needed to service the status modules.

REFERENCES

[1] J. R. Alexander, B. Corker, S. V. Davis, M. T. Heron, A. Oates, “A CAN Based Status Control and Interlock Protection System for the SRS”, International Conference on Accelerator and Large Experimental Physics control Systems, 1999, Trieste, Italy.

[2] http://www.aps.anl.gov/epics/

[3] B.Corker, M.T.Heron, B.G.Martlew, W.R. Rawlinson, “A New Control System for Beamline 5D on the SRS at Daresbury. Sixth European Particle Accelerator Conference (EPAC'98).

[4] http://www.4gls.ac.uk/