Space-Shuttle Emulator Software
Lyndon B. Johnson Space Center, Houston, Texas

A package of software has been developed to execute a raw binary image of the space shuttle flight software for simulation of the computational effects of operation of space shuttle avionics. This software can be run on inexpensive computer workstations. Heretofore, it was necessary to use real flight computers to perform such tests and simulations. The package includes a program that emulates the space shuttle orbiter general-purpose computer (see figure) [consisting of a central processing unit (CPU), input/output processor (IOP), master sequence controller, and bus-control elements]; an emulator of the orbiter display electronics unit and models of the associated cathode-ray tubes, keyboards, and switch controls; computational models of the data-bus network; computational models of the multiplexer-demultiplexer components; an emulation of the pulse-code modulation master unit; an emulation of the payload data interface; a model of the master timing unit; a model of the mass memory unit; and a software component that ensures compatibility of telemetry and command services between the simulated space shuttle avionics and a mission control center. The software package is portable to several host platforms.

This program was written by Scott Arnold, Bill Ashew, Matthew Barry, Agnes Leigh, Scott Mermelstein, James Owens, Dan Payne, John Sollinger, Hiram Thompson, James C. Thompson, Patrick Walter, David Brammel and Steven P. Weismuller of United Space Alliance, LLC; and Ron Aasden, Keith Hurley, and Chris Rahle of Raytheon Co. for Johnson Space Center. Further information is contained in a TSP (see page 1). MSC-23289-1

Soft Real-Time PID Control on a VME Computer
NASA’s Jet Propulsion Laboratory, Pasadena, California

microPID (μPID) is a computer program for real-time proportional + integral + derivative (PID) control of a translation stage in a Fourier-transform ultraviolet spectrometer. μPID implements a PID control loop over a position profile at sampling rate of 8 kHz (sampling period 125 μs). The software runs in a stripped-down Linux operating system on a VersaModule Eurocard (VME) computer operating in real-time priority queue using an embedded controller, a 16-bit digital-to-analog converter (D/A) board, and a laser-positioning board (LPB).

μPID consists of three main parts: (1) VME device-driver routines, (2) software that administers a custom protocol for serial communication with a control computer, and (3) a loop section that obtains the current position from an LPB-driver routine, calculates...