Transmitter Automation

R. M. Gosline
R. F. Systems Development Section

As part of the radio frequency automation demonstration, the transmitter subsystem at DSS 14 was automated to perform a simulated spacecraft mission with pre-mission and post-mission calibrations. Schedule and budgetary restrictions required the use of existing hardware wherever possible. A PDP-11 minicomputer was available and used for the subsystem controller.

I. Introduction

As part of the radio frequency (RF) automation demonstration, the transmitter subsystem at DSS 14 was automated to perform a simulated spacecraft mission with pre-mission and post-mission calibrations. Schedule and budgetary restrictions required the use of existing hardware wherever possible. A PDP-11 minicomputer was available and used for the subsystem controller.

II. Hardware Description

Three junction boxes were installed in the pedestal research and development (R&D) control room near the R&D remote control console (RCC). All monitor and control lines were switched with three rotary switches, enabling the station to return easily to the normal configuration. Logic level converters were used to interface the 28-volt transmitter control logic to the standard 5-volt transistor–transistor logic (TTL) used by the PDP-11. The PDP-11 monitored 64 input lines, 10 analog channels, and provided 19 command lines. The 14-line interface was converted to a serial bit pattern requiring only two twisted pairs for the 370-meter (1200-foot) communication cable to the demonstration controllers located in the DSS 14 control room. A converter at the other end converted back to standard 14 lines. The PDP-11 was equipped with an analog-to-digital converter and a 16-channel multiplexer. The memory size was 16,384 although only 6,000 was available for the control program. A Dectape dual magnetic tape drive was also available.

III. Software Description

The major part of the project involved software development. Top-down structured programming principles were followed, and most of the programming was done in DEC RT-11 BASIC. Even the standard 14-line interface algorithm was implemented in BASIC as a last resort when assembly language routines failed to link string variables properly. Although reliable, only 5 ASCII characters were transferred per second between computers. The program was designed to decode and act on six commands from the demonstration controller as follows (Fig. 1).

INT: The initialize command caused the subsystem to indicate status (manual mode or computer control) to the demonstration controller. If in computer mode, the control power supply, the coolant system, and the filaments were turned on, starting the five-minute filament time delay.

CFG: A configuration command indicated that power and frequency data were following. The received values were checked for proper range and converted for internal use.

CAL: A calibration command would be rejected if INT and CFG were not complete. If accepted and in the waterload position, the crowbar logic was tested, perveance data obtained, and klystron saturated by manipulation of the beam voltage and drive power. Verification of the power meter calibration at the mission power level was checked. Finally, the beam was turned off and the standby state entered.

STB: The standby state could be commanded at any time provided CAL had been completed.

PWR: The power command caused the beam voltage to be turned on to the value specified in CFG provided CAL had been completed. The value of output power was reported to the demonstration controller.

OFF: An off command was only accepted if not in PWR state. An orderly shutdown was performed, allowing time to cool the klystron before shutting off the coolant system. Finally, the computer executes a stop command requiring manual intervention to restart.

All command messages were echoed to the demonstration controller, if possible, otherwise placed on the message output queue to try later. When not performing commands or handling messages, the program entered a monitor loop in which the subsystem parameters were compared to those expected. Anomalies were acted upon, if possible, and reported to the demonstration controller in any case. All analog channels could be corrected for slight drifts without mission interruption. Transmitter interlocks would be reset three times before issuing a diagnostic message requesting manual assistance.

IV. Performance

Many difficulties were encountered with the PDP-11, even though the computer had been reconditioned at the vendor facility. Many RT-11 operating system problems were also discovered for which the vendor could not find solutions.

The second major problem was communication speed due to the software 14-line interface and a polling system instead of interrupts. Long messages could tie up the demonstration controller for minutes while others were polling to see if the controller was ready to listen.

In spite of these difficulties, an automated system was demonstrated, although the klystron saturation algorithm was not completely verified in the demonstration.

V. Conclusions

It should be pointed out that although the automation demonstration provided much data and experience at the system level, few improvements were made within the transmitter subsystem. Instead, the operator was automated at the control panel. There are undoubtedly greater benefits to be obtained by designing certain devices and techniques into the equipment first; then integration into a system demonstration becomes a relatively easy task.

The top-down structured programming technique is a useful method but the rules must be followed rigorously even under schedule pressure to obtain the maximum benefits.

An interrupt system, rather than the polling technique, would greatly improve program execution.

The BASIC language, because of its resident compiler, does not seem to make efficient use of core or machine time. Also, bit manipulations and special device input/output operations are difficult. An assembly language program would seem to offer more advantages as well as preserve the on-line near-real-time correction capability.

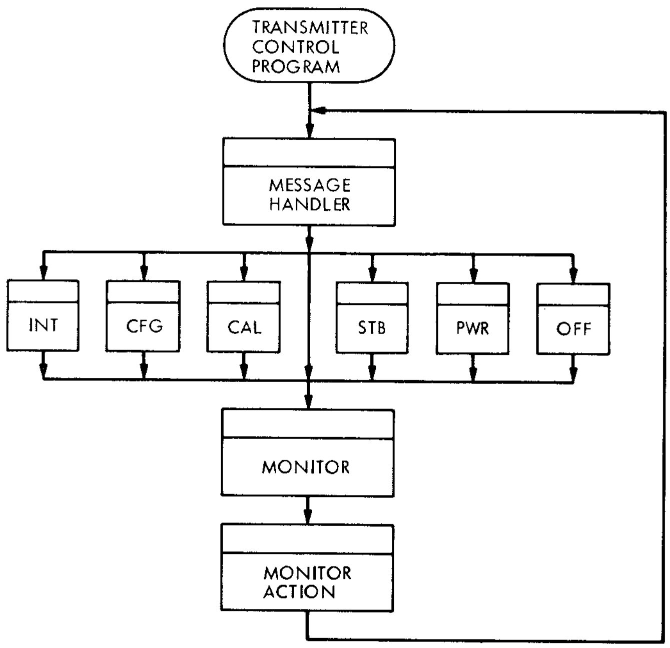

Fig. 1. Transmitter control program top-level flow chart