NASA Technical Memorandum

NASA TM - 103506

HUBBLE SPACE TELESCOPE SIX-BATTERY TEST BED

By J.A. Pajak, J.R. Bush, Jr., and J.R. Lanier, Jr.

Information and Electronic Systems Laboratory
Science and Engineering Directorate

August 1990

National Aeronautics and Space Administration

George C. Marshall Space Flight Center

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM- 103506		

4. Title and Subtitle	5. Report Date
Hubble Space Telescope Six-Battery Test Bed	August 1990
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
JA. Pajak, J.R. Bush, Jr., and J.R. Lanier, Jr.	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812	13. Type of Report and Period Covered Technical Memorandum

12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code
National Aeronautics and Space Administration Washington, DC 20546	

15. Supplementary Notes

Prepared by Information and Electronic Systems Laboratory,
Science and Engineering Directorate

16. Abstract

A test bed for a large space power system breadboard for the Hubble Space Telescope (HST) was designed and built to test the system under simulated orbital conditions. A discussion of the data acquisition and control subsystems designed to provide for continuous 24-hour per day operation and a general overview of the test bed is presented. The data acquisition and control subsystems provided the necessary monitoring and protection to assure safe shutdown with protection of test articles in case of loss of power or equipment failure over the life of the test (up to 5 years).

17. Key Words (Suggested by Author(s))	18. Distribution Statement
space power system testing, HST power system, HST power system test bed, battery testing, nickel-cadmium testing	Unclassified – Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	103	NTIS

For sale by the National Technical Information Service, Springfield, VA 22161-2171

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

TECHNICAL MEMORANDUM

HUBBLE SPACE TELESCOPE SIX-BATTERY TEST BED

BACKGROUND

A power system test bed has been developed to test the Hubble Space Telescope (HST) power system using nickel-cadmium (Ni-Cd) batteries. The test bed provides orbital simulation for charging and discharging the batteries under various operational configurations. Orbital sequencing and data acquisition is accomplished by the data acquisition system (DAS) with solar array simulation (SAS) and load sequencing controlled by the control computer (CC).

The test bed provides space and control for a six-battery power system configuration. The batteries are assembled from Eagle-Picher type 44 Ni-Cd cells, assembled in flight type cases and incorporate battery protection and reconditioning circuits (BPRC).

Test operations are designed for continuous, unattended, operations 24 hours a day, 365 days a year. This type of long term operation (3 to 5 years) requires special considerations for accumulation of data, failure protection, and protection of the test articles and supporting equipment.

PURPOSE

The HST six-battery Ni-Cd test bed was designed to provide a testing facility for the HST power sytem flight configuration. Testing operations were designed to simulate the HST power operations at liftoff, solar array deployment, orbital insertion, and in-orbit operations. The facility also simulates battery environment temperatures. Data collection and subsequent analysis require large volumes of data for trend analysis and anomaly investigation.

SYSTEM CONFIGURATION

A diagram of the HST flight operating system is shown in figure 1. Figure 2 is a block diagram of the test bed. Total duplication of the flight system is, of course, impossible, but every attempt was made to duplicate flight operations on the batteries and their operation in the system. Certain modifications were made to the physical battery configuration to allow insertion in an environmental chamber and for additional nonflight instrumentation, such as cell pressures, cell voltages, and additional temperature measurements. The test bed is located in building 4475, rooms 139 and 140. The test bed configuration design also includes provisions for real-time, real-data transmission to an expert system development test bed (NICBES).

CHAMBER AND BATTERY PHYSICAL CONFIGURATION

The environmental chamber employed in a test bed has a 64 ft^3 (4x4x4 ft) operating space with a conditioned gas inlet at the bottom rear and an exhaust outlet at the top rear. The chamber is purged with dry nitrogen as the environmental medium. The nitrogen reduces the likelihood of condensation during operation and when the chamber must be opened for test bed repairs or reconfiguration of physical devices. The nitrogen is under a slight positive pressure and is vented to the outside of the building. The chamber is easily capable of maintaining a set temperature in the operating range of −10 °C to 30 °C with a 1-kW heating load.

The batteries are divided into two groups of three batteries. Each of the battery groups is mounted on a cold plate. The cold plates have imbedded heaters to allow separate control of the battery group temperatures. The cold plates, with batteries, are stacked vertically in a support frame. Insulation surrounds the battery groups on top, sides, and ends to force most generated heat to be extracted via the cold plates. Baffling exists in the chamber to direct the circulating ambient environment to the cold plates for proper heat transfer. The battery physical arrangement thus provides a thermal path similar to that expected in space operation.

A physical modification to the battery cases has been made to accommodate the individual cell pressure transducers and their attendant cabling and to allow wiring to each cell terminal to obtain cell voltages. This modification is accomplished by using 1-inch spacers between the battery case and lid. The lid is still within the insulation surrounding the groups for thermal integrity.

Figure 3 is a depiction of the chamber physical layout. Figure 4 is a photograph of the installed batteries.

TEST CONTROL AND DATA ACQUISITION

Signals from the DAS control the orbital day/night time of the test bed. Day/night orbital times are synchronized with the DAS 1-min scan rate to preclude data acquisition during the battery charge-discharge transitions, and to allow stabilization of operating parameters between readings. The DAS also performs limit checking on the data to alert to failure conditions of support equipment such as environmental chamber cooling failure, building power failure, SAS equipment failure, and cell-battery failure.

Table 1 lists the types and distribution of data types read by the DAS, while appendices A through C list the channel assignments. The DAS is a system assembled by DSP Inc. utilizing a LSI-11/23 DEC board computer as the processor. The operating system is RT-11 with the operating programs written in BASIC for ease of modification and interactive operator control. Data scanning is accomplished using 16-channel, flying-capacitor, reed-relay multiplexors. Network completion circuits are also provided for resistance measurements of thermistor temperature circuits.

Figure 5 is a block diagram of the DAS system. A typical control circuit for a single battery is shown in figure 6.

The DAS also incorporates a switch panel to provide operator control of scan data printouts, recycling of orbital position to the beginning of day or night (start of new orbit), and forcing discharge during day or charge during night phase.

To accommodate the accumulation and retention of large amounts of data (370 items per scan), an external "AT" class computer has been attached to a RS232 port. This computer incorporates an optical disk of the "write once-read many" (WORM) technology. Two hundred megabytes of storage are available per disk. This represents about 3 weeks of data. This same data is also routed to an expert system test bed for real-time development of artificial intelligence (AI) technologies as applied to large space power systems.

The DAS also has two printer outputs. One printer is used for listing orbital summary data (fig. 11) and the second printer lists failure conditions and scan data requests (fig. 12).

Appendix D is the control and data acquisition program for the LSI-11/23 with comments added for clarity. Also included are the DAS operating instructions.

SOLAR ARRAY SIMULATORS

The HST orbital configuration consists of 20 solar panels divided into groups of 3 panels for each battery, with the remaining 2 panels connected directly to bus C. For test simulation, two power supplies per battery are employed. One power supply (type 1) of the pair simulates a single solar panel, the second supply (type 2) simulates the output of two solar panels. The SAS control circuits can thus simulate the power delivered by type 1, 2, or 3 panels, as determined by the charge current controllers (CCC). A type 2 supply simulates the two solar power arrays (SPA's) connected to bus C.

LOAD BUSES

The HST three-flight load buses are replicated by three programmable load banks. Each load bank is independently controlled by the CC to perform in the predicted orbital configuration, including the failure SAFEMODE.

CHARGE CURRENT CONTROLLER

The charge voltage of each battery is controlled by a CCC. Each CCC controls the number and battery cut-off voltages of the SPA's (SAS) charging each battery. The cutoff voltages are temperature compensated to reduce charging stress at higher temperatures.

CONTROL COMPUTER

The control computer is an inhouse designed and fabricated microprocessor system. The CC has an operator interface for entry of desired orbital operating parameters. A TI 9900 chip is used as the central processing unit (CPU) with series supporting chips and 2k bytes of EPROM memory. The CC provides for independent SAS control (simulating HST orientation and SPA degradation), load bank control (simulating various load conditions), and monitoring of CCC operation, battery temperature, battery voltage, and DAS operation. The CC circuits also provide failure monitoring of the SAS and CCC. The CC incorporates a 16-channel multiplexor and 12-bit analog/digital (A/D) converter for analog data monitoring. CC output control is accomplished via two 16-bit I/O chips with relay driver interface. The CC logic and board circuitry are shown in figures 7 through 10.

The CC operating program is programmed into the EPROM memory. Appendix E is a listing of the two programs for CC operation and operator interface.

TEST SHUTDOWN AND FAILURE PROTECTION

The number of individual subsystems and integrated operation of the test bed requires several levels of failure protection and test shutdown. Since the test bed is earthbound and dependent on utility power for operations, an uninterruptable power source supplies the operating power for the DAS and CC. Since the test bed is designed for continuous, unattended, operation all circuits have FAIL-SAFE provisions, and the DAS and CC are connected to an autodialer circuit to notify test personnel of a failure night or day.

Protection of the test articles (batteries) and the test bed equipment is an important consideration, therefore the conditions listed in table 2 are considered faults and initiate the indicated actions. Test shutdown for failures requires operator intervention for restart, even if the fault self corrects. This is a safety precaution to verify that the fault cause is found and properly cleared or corrected.

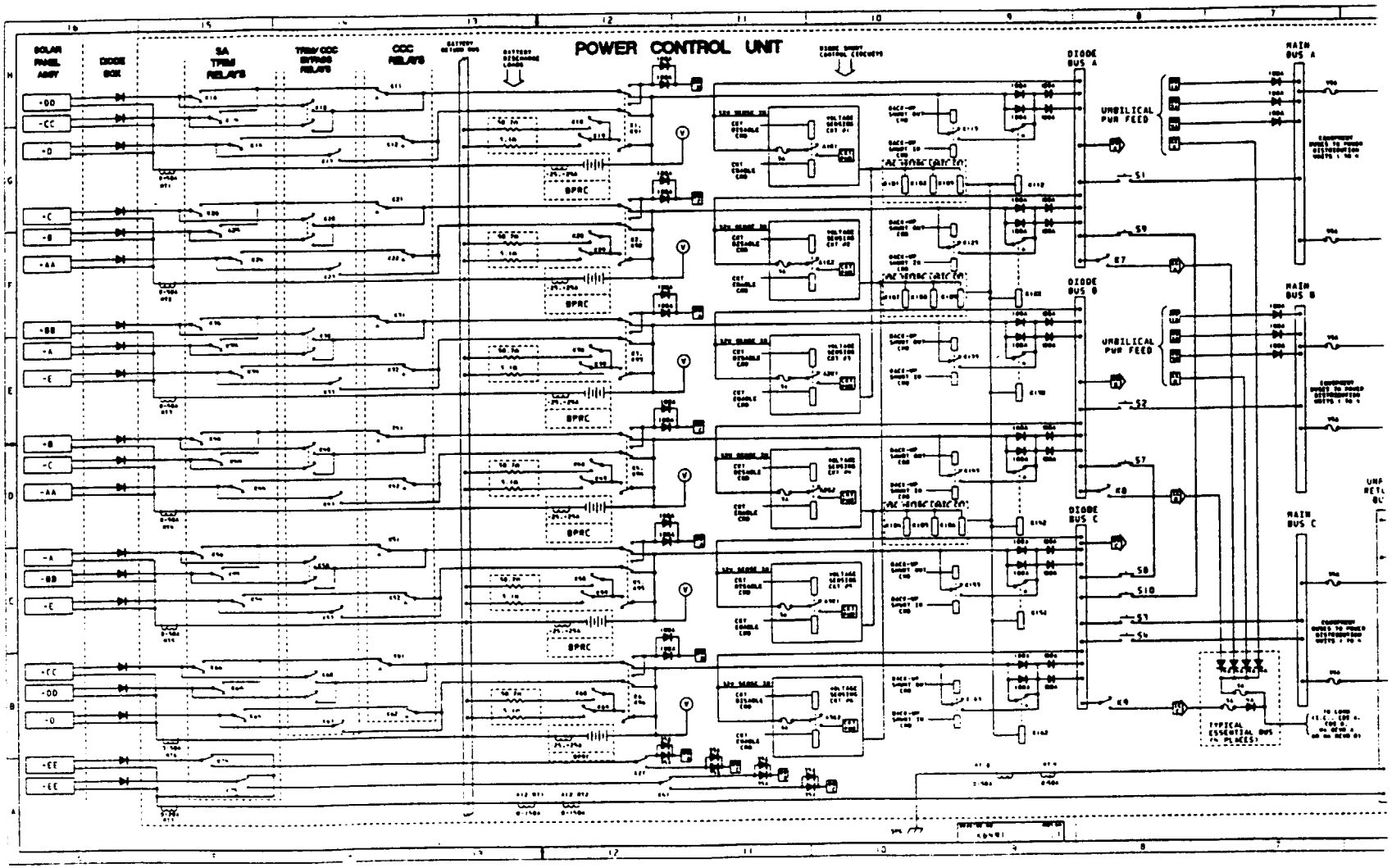

Figure 1. HST power system.

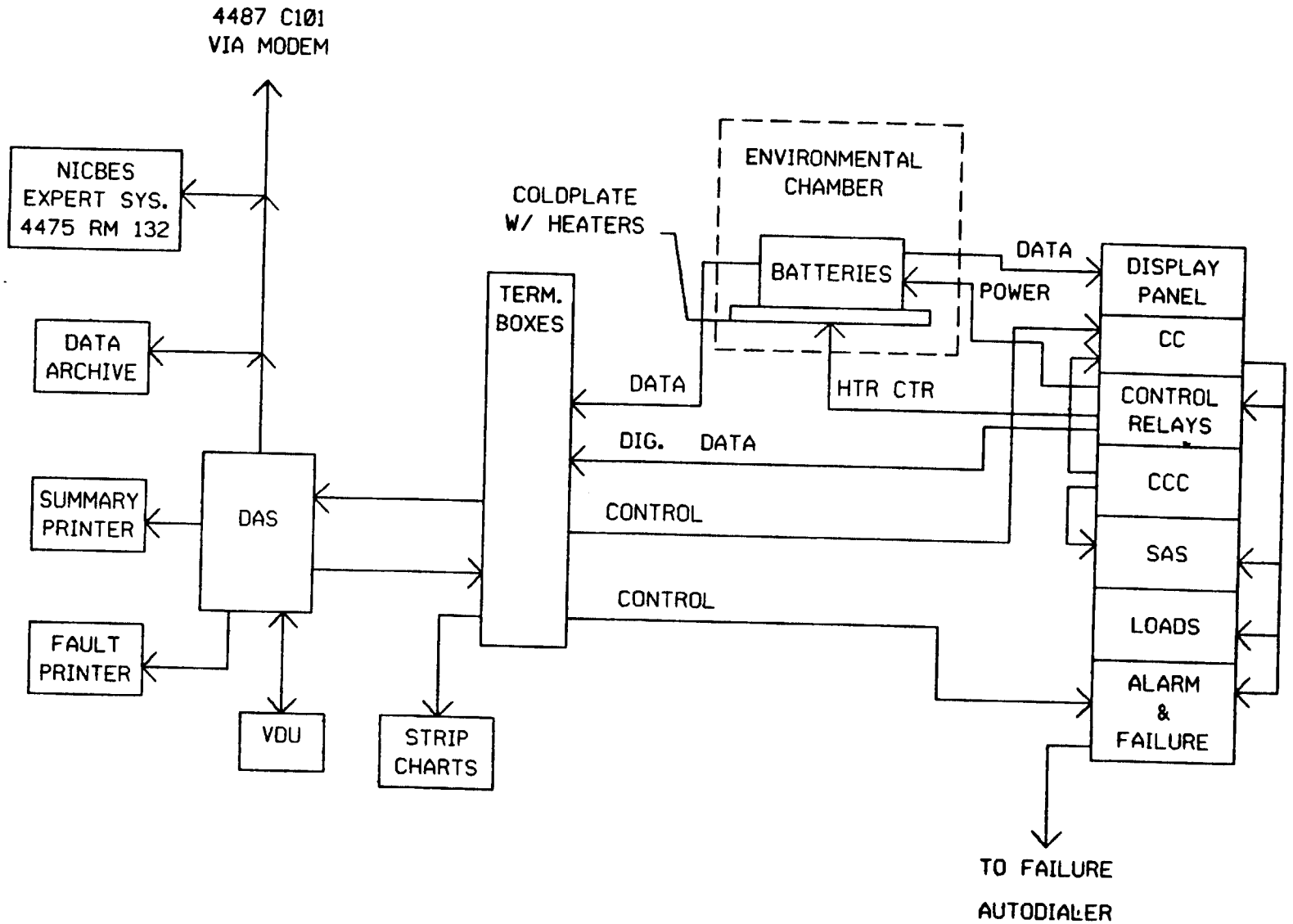

Figure 2. Test bed block diagram.

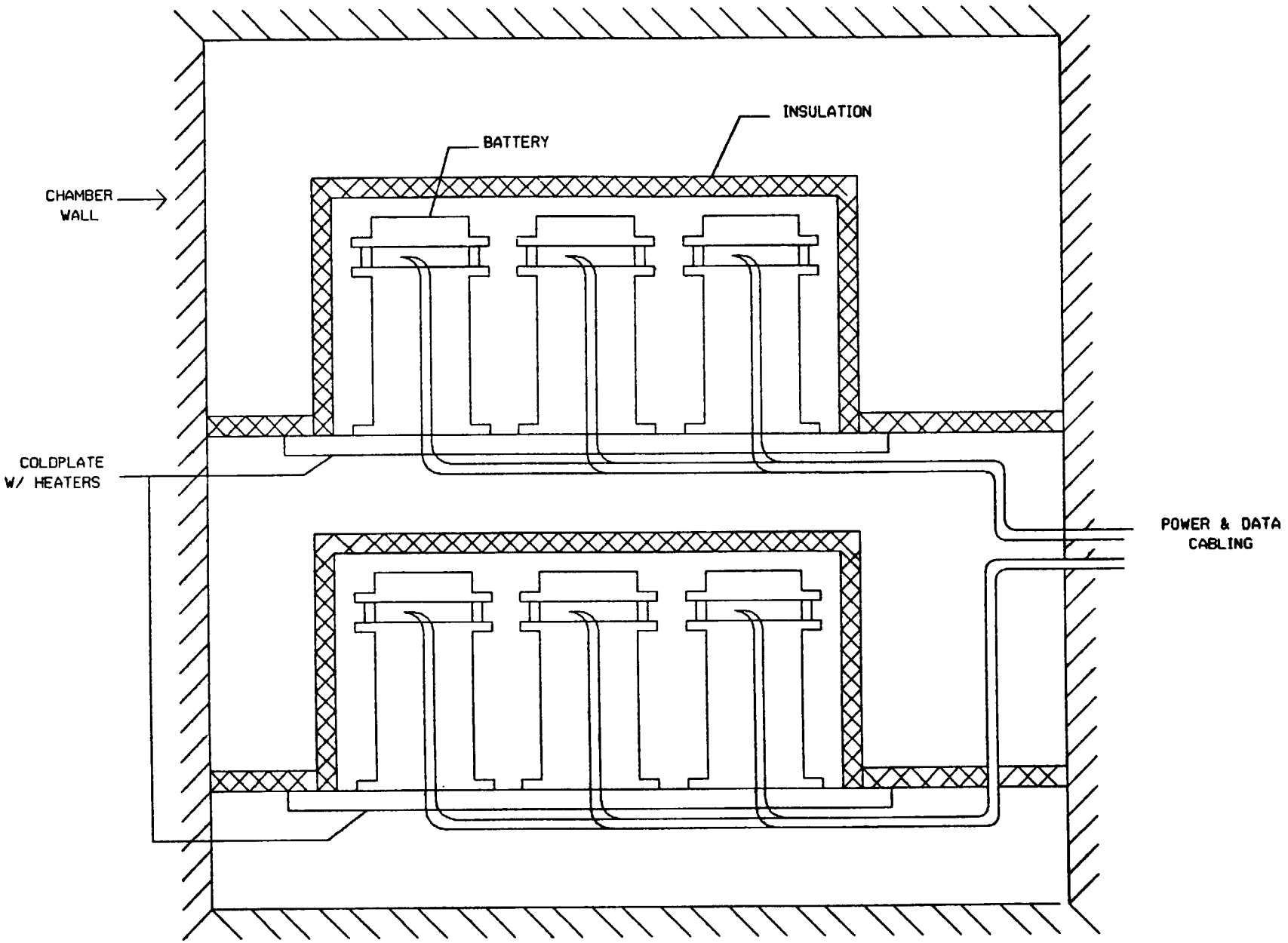

Figure 3. Chamber physical layout.

Figure 4. Installed batteries.

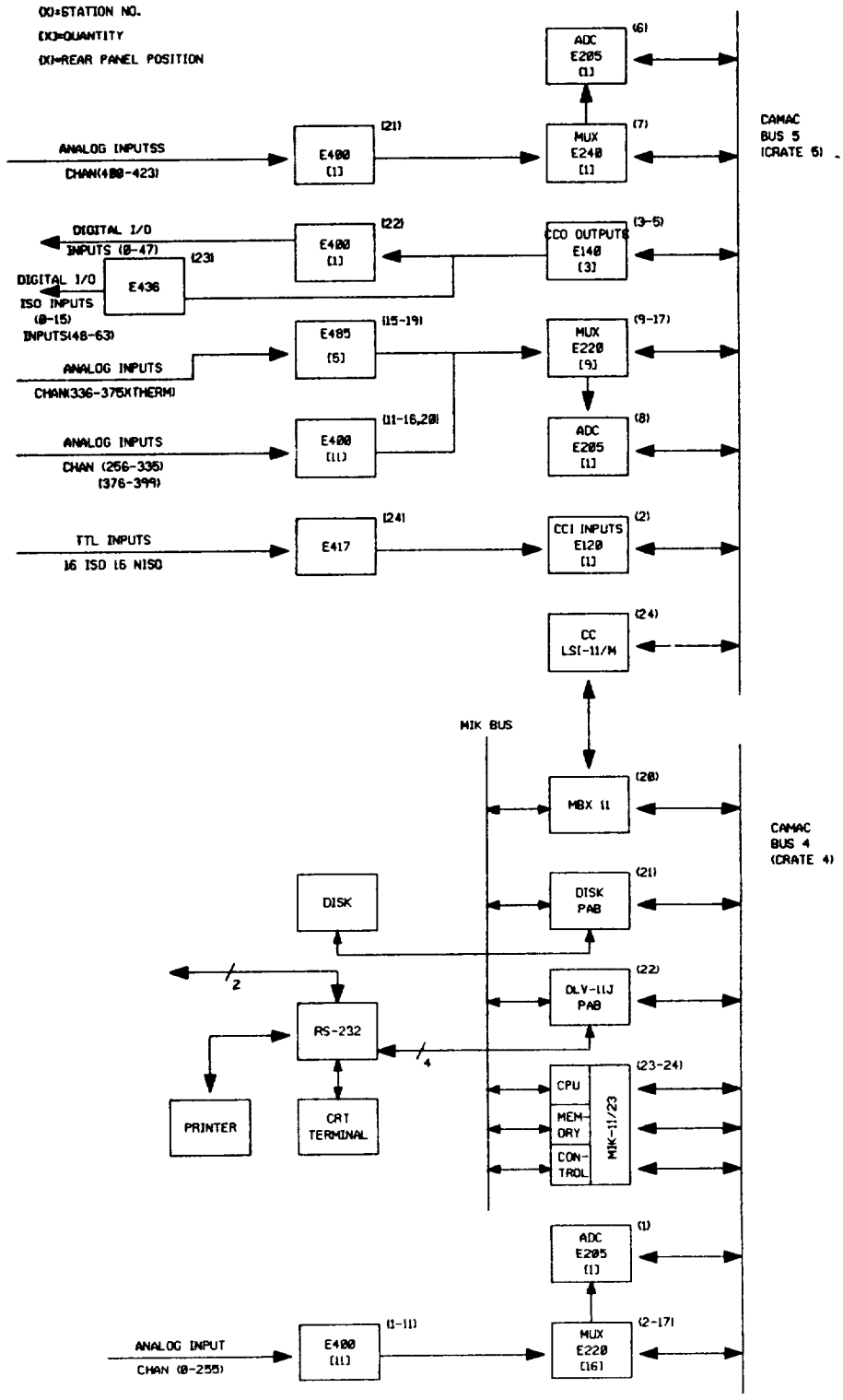

Figure 5. DAS system block diagram.

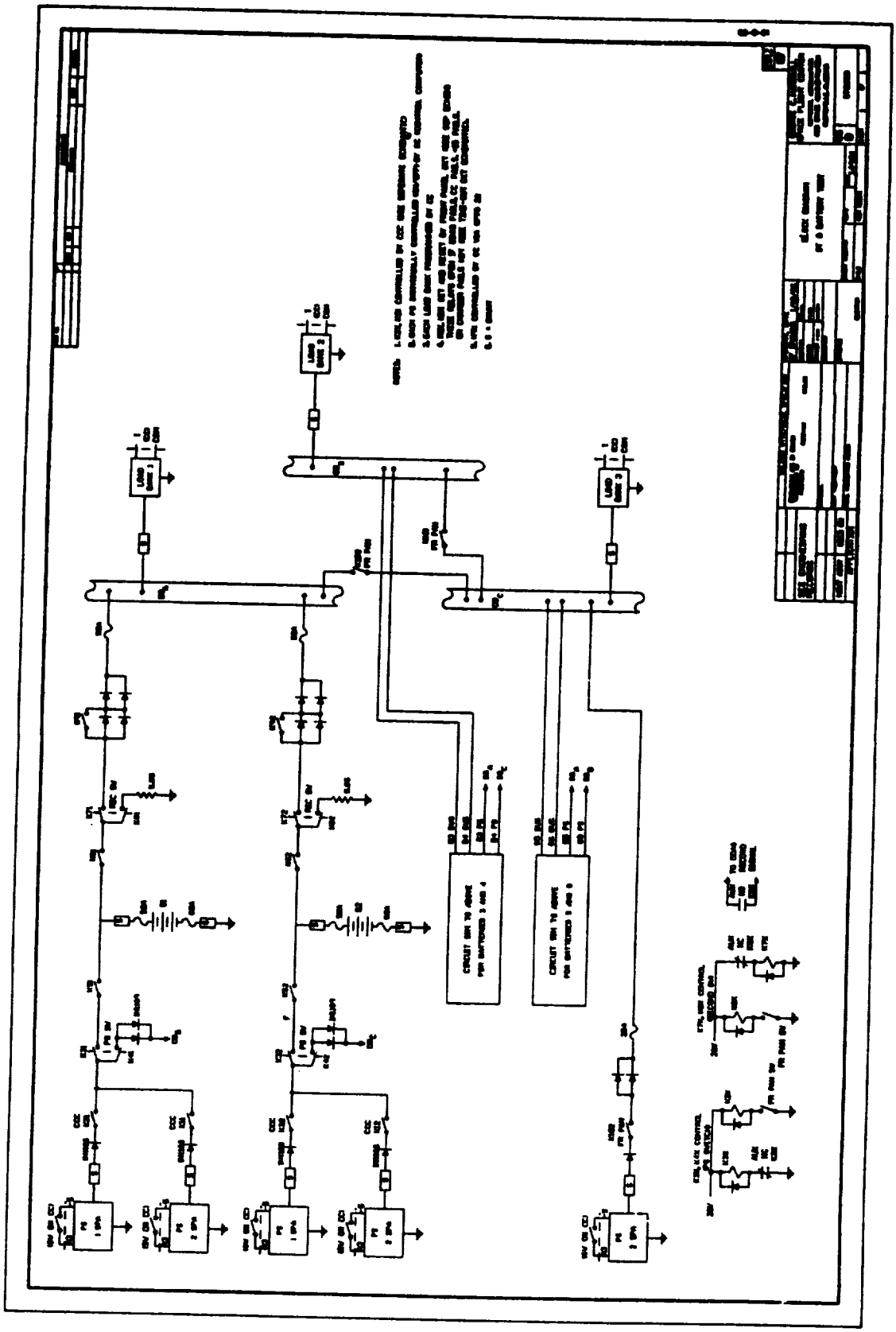

Figure 6. Single battery control circuit.

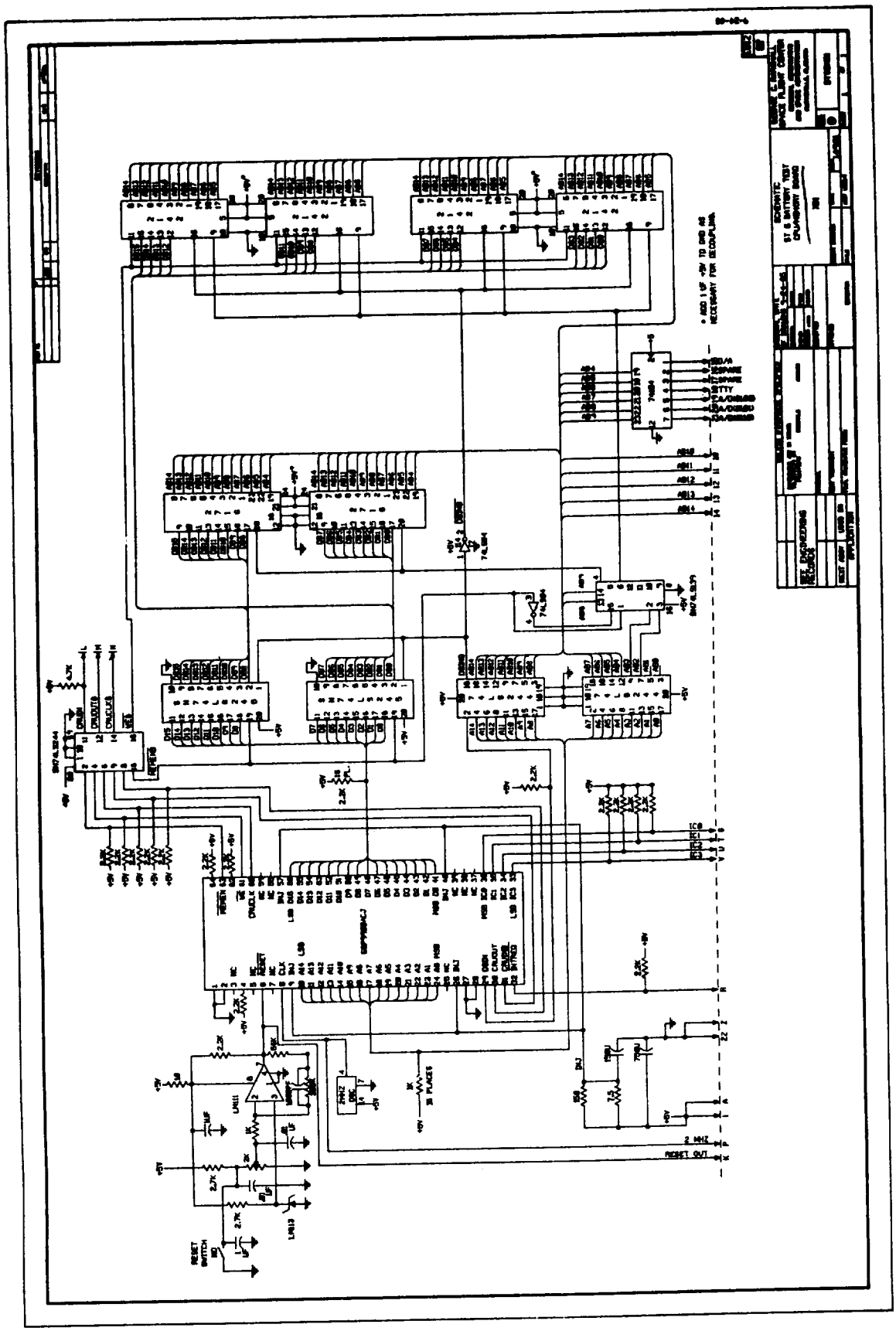

Figure 7. CPU/memory board.

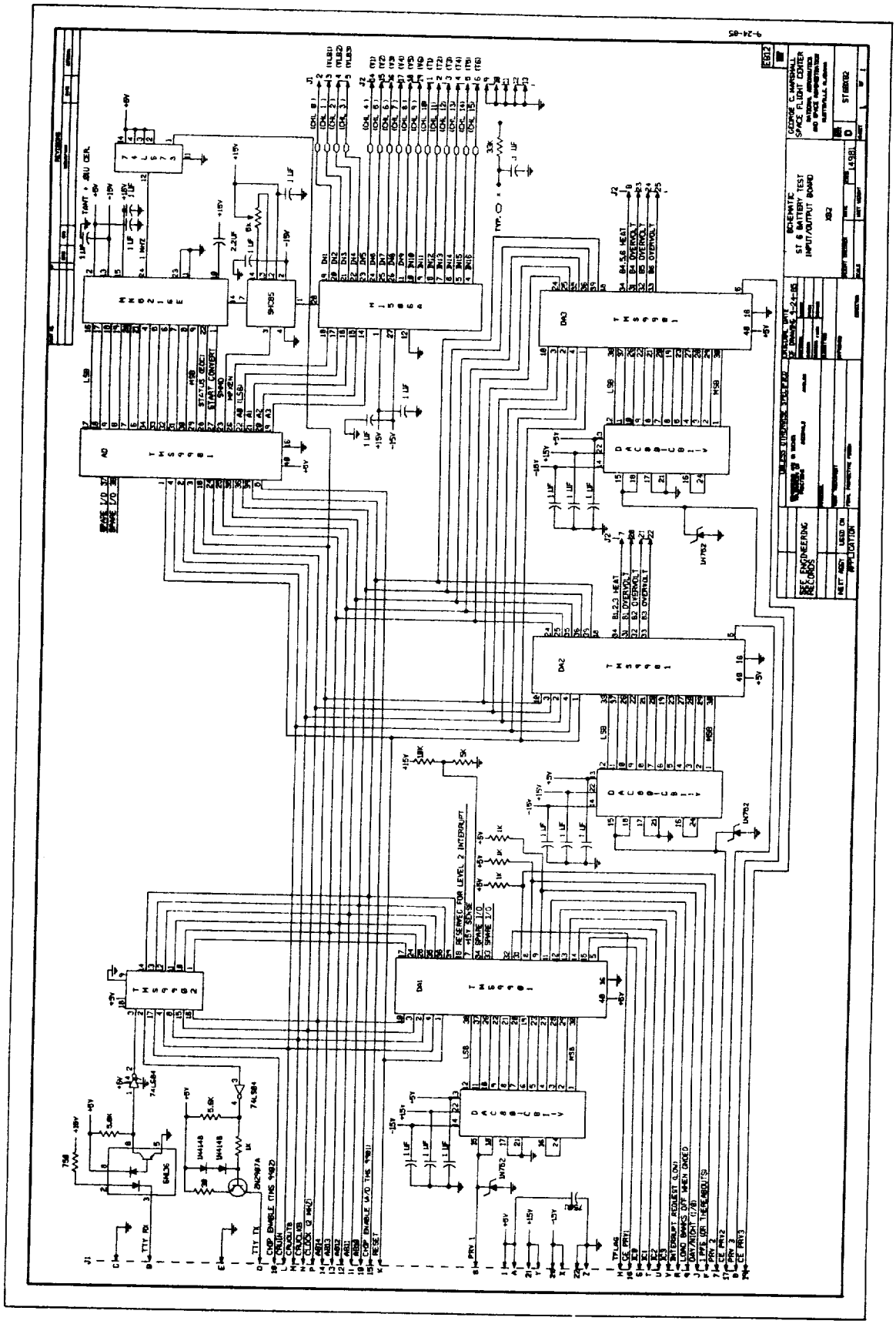

Figure 8. Input/output board.

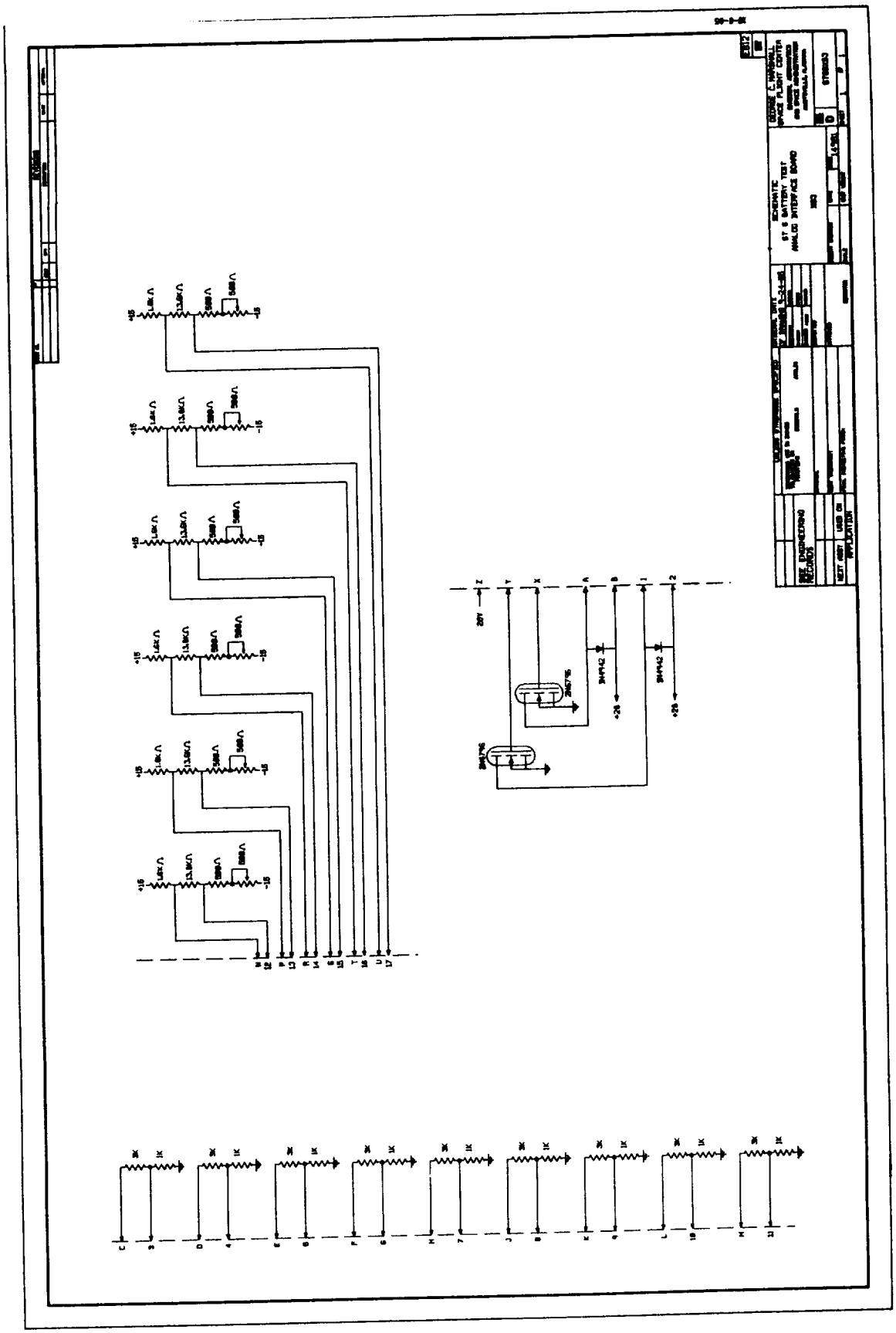

Figure 9. Analog interface board.

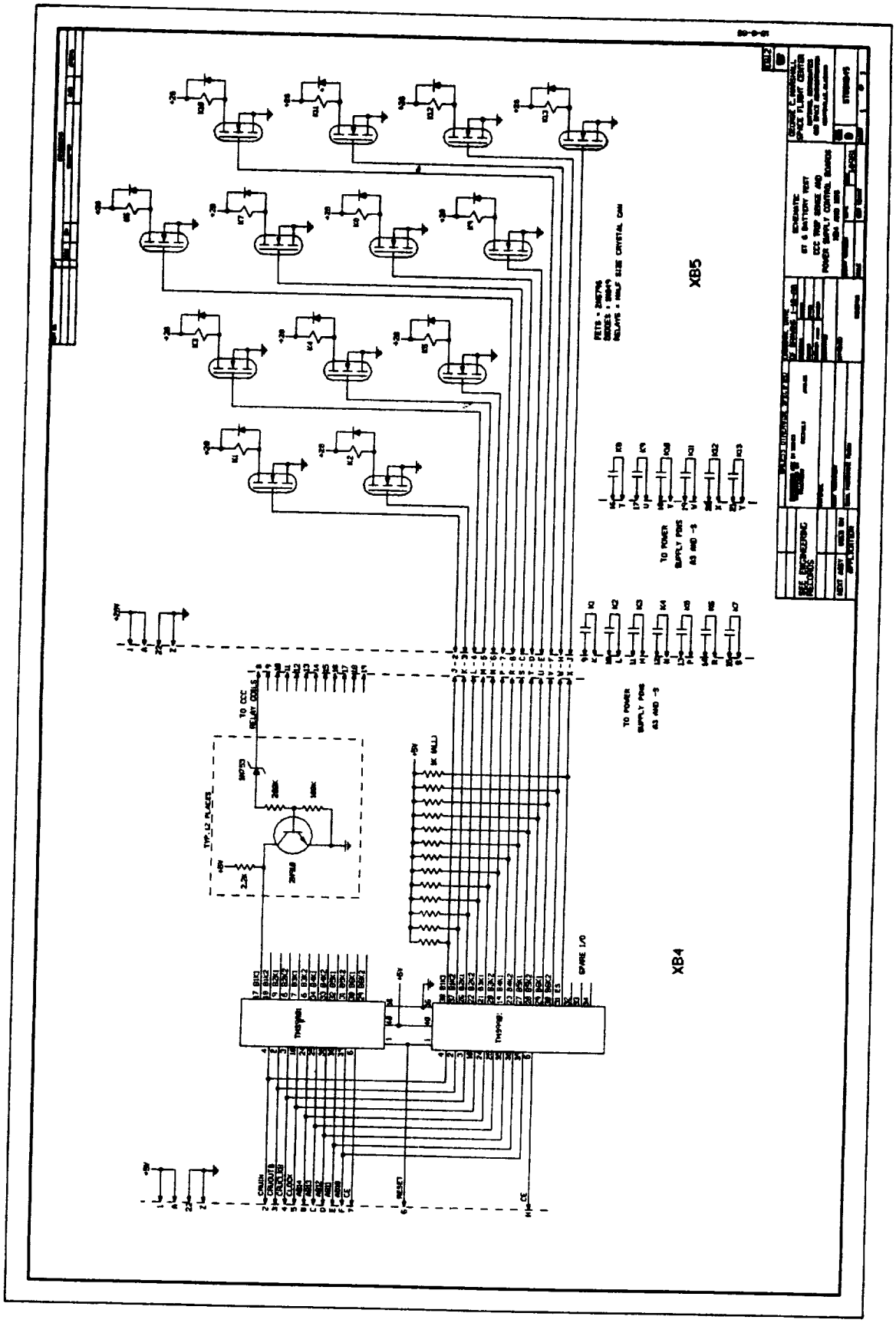

Figure 10. CCC trip sense and power supply control boards XB4 and XB5.

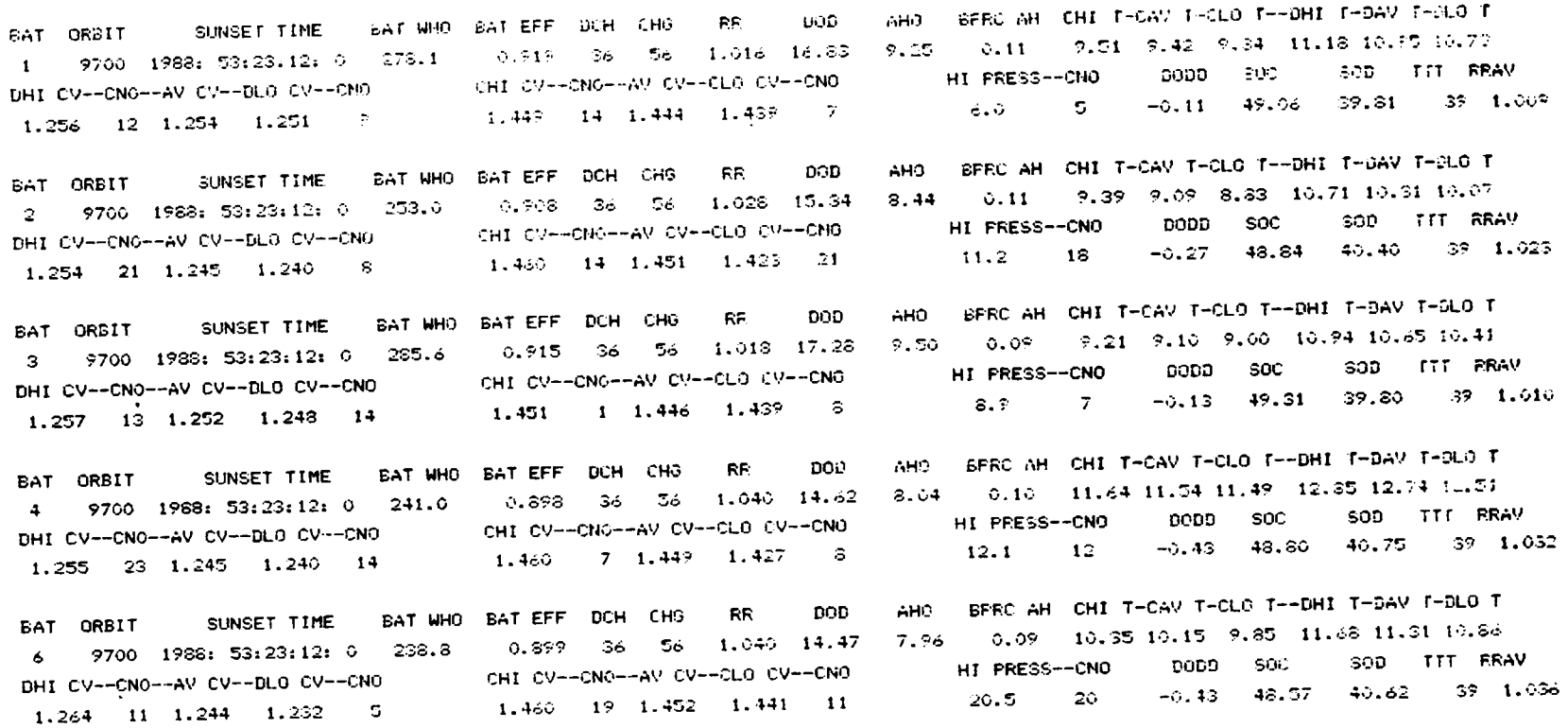

Figure 11. Orbital summary data.

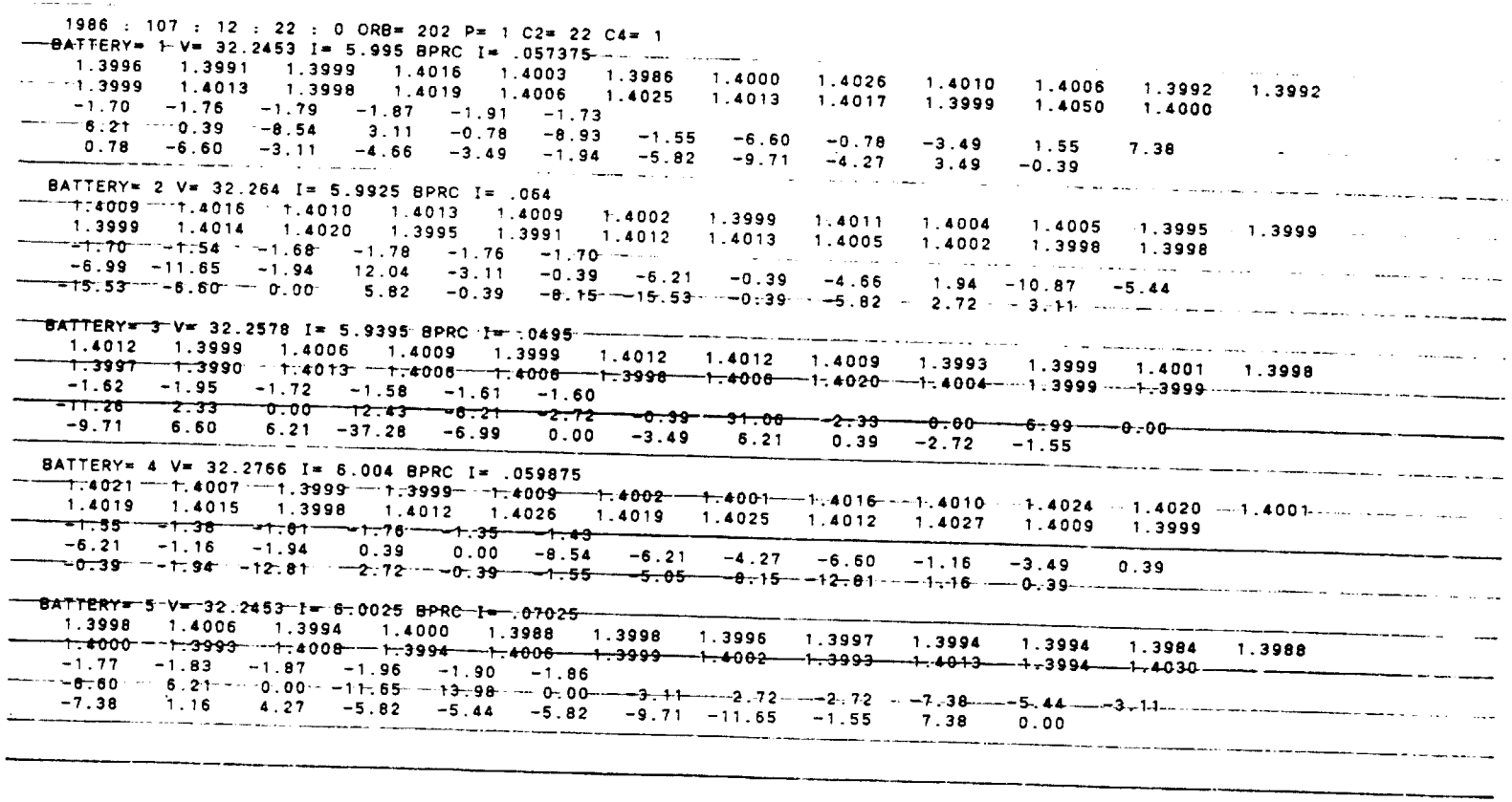

Figure 12. Failure conditions and scan data requests.

Table 1. Data types read by the DAS.

DAS channels	type of signal	Quan.	Use
ANALOG			
0-255	0-+/- 2.4 v	256	cell volt. and cell press.
256-335	0-+/- 2.4 v	80	same as above
336-375	0-100k ohms	40	theristor temperature
376-399	0-200 v	24	battery and load volts.
400-415	0-+/- 100 mv	16	battery and SAS curr.
DIGITAL			
0-63	TTL level	64	cc contol
0-15	switches	16	alarm and level control
0-15	28 v level	16	operator control panel
16-31	TTL level	16	battery status

Table 2. Fault conditions.

Fault	Action
Environmental chamber temp. too high	Shutdown
Environmental chamber temp. too low	Shutdown
SAS current too high	Alarm
DAS not scanning	Shutdown
Cell voltage too high	Shutdown
Cell voltage too low	Shutdown
Bus current too high	Shutdown
SAS curr. too low at start of charge	Alarm
Utility power failure	Alarm

Appendix A
ANALOG SIGNALS

Channel	Use	Voltage (Volts) (+ or -)
0		2.5
1	Batt 1 / Cell 1	
2	Batt 1 / Cell 2	
3	Batt 1 / Cell 3	
4	Batt 1 / Cell 4	
5	Batt 1 / Cell 5	
6	Batt 1 / Cell 6	
7	Batt 1 / Cell 7	
8	Batt 1 / Cell 8	
9	Batt 1 / Cell 9	
10	Batt 1 / Cell 10	
11	Batt 1 / Cell 11	
12	Batt 1 / Cell 12	
13	Batt 1 / Cell 13	
14	Batt 1 / Cell 14	
15	Batt 1 / Cell 15	
16	Batt 1 / Cell 16	
17	Batt 1 / Cell 17	
18	Batt 1 / Cell 18	
19	Batt 1 / Cell 19	
20	Batt 1 / Cell 20	
21	Batt 1 / Cell 21	
22	Batt 1 / Cell 22	
23	Batt 1 / Cell 23	
24	Batt 2 / Cell 1	
25	Batt 2 / Cell 2	
26	Batt 2 / Cell 3	
27	Batt 2 / Cell 4	
28	Batt 2 / Cell 5	
29	Batt 2 / Cell 6	
30	Batt 2 / Cell 7	
31	Batt 2 / Cell 8	
32	Batt 2 / Cell 9	
33	Batt 2 / Cell 10	
34	Batt 2 / Cell 11	
35	Batt 2 / Cell 12	
36	Batt 2 / Cell 13	
37	Batt 2 / Cell 14	
38	Batt 2 / Cell 15	
39	Batt 2 / Cell 16	
40	Batt 2 / Cell 17	
41	Batt 2 / Cell 18	
42	Batt 2 / Cell 19	
43	Batt 2 / Cell 20	
44	Batt 2 / Cell 21	
45	Batt 2 / Cell 22	
46	Batt 2 / Cell 23	2.5

Appendix A cont.

Channel	Use	Voltage (Volts) (+ or -)
47	Batt 3 / Cell 1	2.5
48	Batt 3 / Cell 2	
49	Batt 3 / Cell 3	
50	Batt 3 / Cell 4	
51	Batt 3 / Cell 5	
52	Batt 3 / Cell 6	
53	Batt 3 / Cell 7	
54	Batt 3 / Cell 8	
55	Batt 3 / Cell 9	
56	Batt 3 / Cell 10	
57	Batt 3 / Cell 11	
58	Batt 3 / Cell 12	
59	Batt 3 / Cell 13	
60	Batt 3 / Cell 14	
61	Batt 3 / Cell 15	
62	Batt 3 / Cell 16	
63	Batt 3 / Cell 17	
64	Batt 3 / Cell 18	
65	Batt 3 / Cell 19	
66	Batt 3 / Cell 20	
67	Batt 3 / Cell 21	
68	Batt 3 / Cell 22	
69	Batt 3 / Cell 23	
70	Batt 4 / Cell 1	
71	Batt 4 / Cell 2	
72	Batt 4 / Cell 3	
73	Batt 4 / Cell 4	
74	Batt 4 / Cell 5	
75	Batt 4 / Cell 6	
76	Batt 4 / Cell 7	
77	Batt 4 / Cell 8	
78	Batt 4 / Cell 9	
79	Batt 4 / Cell 10	
80	Batt 4 / Cell 11	
81	Batt 4 / Cell 12	
82	Batt 4 / Cell 13	
83	Batt 4 / Cell 14	
84	Batt 4 / Cell 15	
85	Batt 4 / Cell 16	
86	Batt 4 / Cell 17	
87	Batt 4 / Cell 18	
88	Batt 4 / Cell 19	
89	Batt 4 / Cell 20	
90	Batt 4 / Cell 21	
91	Batt 4 / Cell 22	
92	Batt 4 / Cell 23	2.5

Appendix A cont.

Channel	Use	Voltage (Volts) (+ or -)
93	Batt 5 / Cell 1	2.5
94	Batt 5 / Cell 2	
95	Batt 5 / Cell 3	
96	Batt 5 / Cell 4	
97	Batt 5 / Cell 5	
98	Batt 5 / Cell 6	
99	Batt 5 / Cell 7	
100	Batt 5 / Cell 8	
101	Batt 5 / Cell 9	
102	Batt 5 / Cell 10	
103	Batt 5 / Cell 11	
104	Batt 5 / Cell 12	
105	Batt 5 / Cell 13	
106	Batt 5 / Cell 14	
107	Batt 5 / Cell 15	
108	Batt 5 / Cell 16	
109	Batt 5 / Cell 17	
110	Batt 5 / Cell 18	
111	Batt 5 / Cell 19	
112	Batt 5 / Cell 20	
113	Batt 5 / Cell 21	
114	Batt 5 / Cell 22	
115	Batt 5 / Cell 23	
116	Batt 6 / Cell 1	
117	Batt 6 / Cell 2	
118	Batt 6 / Cell 3	
119	Batt 6 / Cell 4	
120	Batt 6 / Cell 5	
121	Batt 6 / Cell 6	
122	Batt 6 / Cell 7	
123	Batt 6 / Cell 8	
124	Batt 6 / Cell 9	
125	Batt 6 / Cell 10	
126	Batt 6 / Cell 11	
127	Batt 6 / Cell 12	
128	Batt 6 / Cell 13	
129	Batt 6 / Cell 14	
130	Batt 6 / Cell 15	
131	Batt 6 / Cell 16	
132	Batt 6 / Cell 17	
133	Batt 6 / Cell 18	
134	Batt 6 / Cell 19	
135	Batt 6 / Cell 20	
136	Batt 6 / Cell 21	
137	Batt 6 / Cell 22	2.5

Appendix A cont.

Channel	Use	Voltage (Volts) (+ or -)
138	Batt 6 / Cell 23	2.5
139	Not Used	
140	Not Used	
141	Batt 1 / Cell 1 Press	
142	Batt 1 / Cell 2 Press	
143	Batt 1 / Cell 3 Press	
144	Batt 1 / Cell 4 Press	
145	Batt 1 / Cell 5 Press	
146	Batt 1 / Cell 6 Press	
147	Batt 1 / Cell 7 Press	
148	Batt 1 / Cell 8 Press	
149	Batt 1 / Cell 9 Press	
150	Batt 1 / Cell 10 Press	
151	Batt 1 / Cell 11 Press	
152	Batt 1 / Cell 12 Press	
153	Batt 1 / Cell 13 Press	
154	Batt 1 / Cell 14 Press	
155	Batt 1 / Cell 15 Press	
156	Batt 1 / Cell 16 Press	
157	Batt 1 / Cell 17 Press	
158	Batt 1 / Cell 18 Press	
159	Batt 1 / Cell 19 Press	
160	Batt 1 / Cell 20 Press	
161	Batt 1 / Cell 21 Press	
162	Batt 1 / Cell 22 Press	
163	Batt 1 / Cell 23 Press	
164	Batt 2 / Cell 1 Press	
165	Batt 2 / Cell 2 Press	
166	Batt 2 / Cell 3 Press	
167	Batt 2 / Cell 4 Press	
168	Batt 2 / Cell 5 Press	
169	Batt 2 / Cell 6 Press	
170	Batt 2 / Cell 7 Press	
171	Batt 2 / Cell 8 Press	
172	Batt 2 / Cell 9 Press	
173	Batt 2 / Cell 10 Press	
174	Batt 2 / Cell 11 Press	
175	Batt 2 / Cell 12 Press	
176	Batt 2 / Cell 13 Press	
177	Batt 2 / Cell 14 Press	
178	Batt 2 / Cell 15 Press	
179	Batt 2 / Cell 16 Press	
180	Batt 2 / Cell 17 Press	
181	Batt 2 / Cell 18 Press	
182	Batt 2 / Cell 19 Press	
183	Batt 2 / Cell 20 Press	2.5

Appendix A cont.

Channel	Use	Voltage (Volts) (+ or -)
184	Batt 2 / Cell 21 Press	2.5
185	Batt 2 / Cell 22 Press	
186	Batt 2 / Cell 23 Press	
187	Batt 3 / Cell 1 Press	
188	Batt 3 / Cell 2 Press	
189	Batt 3 / Cell 3 Press	
190	Batt 3 / Cell 4 Press	
191	Batt 3 / Cell 5 Press	
192	Batt 3 / Cell 6 Press	
193	Batt 3 / Cell 7 Press	
194	Batt 3 / Cell 8 Press	
195	Batt 3 / Cell 9 Press	
196	Batt 3 / Cell 10 Press	
197	Batt 3 / Cell 11 Press	
198	Batt 3 / Cell 12 Press	
199	Batt 3 / Cell 13 Press	
200	Batt 3 / Cell 14 Press	
201	Batt 3 / Cell 15 Press	
202	Batt 3 / Cell 16 Press	
203	Batt 3 / Cell 17 Press	
204	Batt 3 / Cell 18 Press	
205	Batt 3 / Cell 19 Press	
206	Batt 3 / Cell 20 Press	
207	Batt 3 / Cell 21 Press	
208	Batt 3 / Cell 22 Press	
209	Batt 3 / Cell 23 Press	
210	Batt 4 / Cell 1 Press	
211	Batt 4 / Cell 2 Press	
212	Batt 4 / Cell 3 Press	
213	Batt 4 / Cell 4 Press	
214	Batt 4 / Cell 5 Press	
215	Batt 4 / Cell 6 Press	
216	Batt 4 / Cell 7 Press	
217	Batt 4 / Cell 8 Press	
218	Batt 4 / Cell 9 Press	
219	Batt 4 / Cell 10 Press	
220	Batt 4 / Cell 11 Press	
221	Batt 4 / Cell 12 Press	
222	Batt 4 / Cell 13 Press	
223	Batt 4 / Cell 14 Press	
224	Batt 4 / Cell 15 Press	
225	Batt 4 / Cell 16 Press	
226	Batt 4 / Cell 17 Press	
227	Batt 4 / Cell 18 Press	
228	Batt 4 / Cell 19 Press	
229	Batt 4 / Cell 20 Press	2.5

Appendix A cont.

Channel	Use	Voltage (Volts) (+ or -)
230	Batt 4 / Cell 21 Press	2.5
231	Batt 4 / Cell 22 Press	
232	Batt 4 / Cell 23 Press	
233	Batt 5 / Cell 1 Press	
234	Batt 5 / Cell 2 Press	
235	Batt 5 / Cell 3 Press	
236	Batt 5 / Cell 4 Press	
237	Batt 5 / Cell 5 Press	
238	Batt 5 / Cell 6 Press	
239	Batt 5 / Cell 7 Press	
240	Batt 5 / Cell 8 Press	
241	Batt 5 / Cell 9 Press	
242	Batt 5 / Cell 10 Press	
243	Batt 5 / Cell 11 Press	
244	Batt 5 / Cell 12 Press	
245	Batt 5 / Cell 13 Press	
246	Batt 5 / Cell 14 Press	
247	Batt 5 / Cell 15 Press	
248	Batt 5 / Cell 16 Press	
249	Batt 5 / Cell 17 Press	
250	Batt 5 / Cell 18 Press	
251	Batt 5 / Cell 19 Press	
252	Batt 5 / Cell 20 Press	
253	Batt 5 / Cell 21 Press	
254	Batt 5 / Cell 22 Press	
255	Batt 5 / Cell 23 Press	
256	Batt 6 / Cell 1 Press	
257	Batt 6 / Cell 2 Press	
258	Batt 6 / Cell 3 Press	
259	Batt 6 / Cell 4 Press	
260	Batt 6 / Cell 5 Press	
261	Batt 6 / Cell 6 Press	
262	Batt 6 / Cell 7 Press	
263	Batt 6 / Cell 8 Press	
264	Batt 6 / Cell 9 Press	
265	Batt 6 / Cell 10 Press	
266	Batt 6 / Cell 11 Press	
267	Batt 6 / Cell 12 Press	
268	Batt 6 / Cell 13 Press	
269	Batt 6 / Cell 14 Press	
270	Batt 6 / Cell 15 Press	
271	Batt 6 / Cell 16 Press	
272	Batt 6 / Cell 17 Press	
273	Batt 6 / Cell 18 Press	
274	Batt 6 / Cell 19 Press	
275	Batt 6 / Cell 20 Press	2.5

Appendix A cont.

Channel	Use	Voltage (Volts) (+ or -)
276	Batt 6 / Cell 21 Press	2.5
277	Batt 6 / Cell 22 Press	
278	Batt 6 / Cell 23 Press	
279	Not used	
280		
281		
282		
283		
284		
285		
286		
287		
288		
289		
290		
291		
292		
293		
294		
295		
296		
297		
298		
299		
300		
301		
302		
303		
304		
305		
306		
307		
308		
309		
310		
311		
312		
313		
314		
315		
316	Not Used	2.5

Appendix A cont.

Channel	Use	Voltage (Volts) (+ or -)
317	Not Used	2.5
318	Not Used	
319	Not Used	
320	BUS 1 CURR	
321	BUS 2 CURR	
322	BUS 3 CURR	
323	SAS 1 CURR	
324	SAS 2 CURR	
325	SAS 3 CURR	
326	SAS 4 CURR	
327	SAS 5 CURR	
328	SAS 6 CURR	
329	SAS 7 CURR	
330	SAS 8 CURR	
331	SAS 9 CURR	
332	SAS 10 CURR	
333	SAS 11 CURR	
334	SAS 12 CURR	
335	SAS 13 CURR	2.5
336	Batt 1 / Temp 1	THERM
337	Batt 1 / Temp 2	
338	Batt 1 / Temp 3	
339	Batt 1 / Temp 4	
340	Batt 1 / Temp 5	
341	Batt 1 / Temp 6	
342	Batt 2 / Temp 1	
343	Batt 2 / Temp 2	
344	Batt 2 / Temp 3	
345	Batt 2 / Temp 4	
346	Batt 2 / Temp 5	
347	Batt 2 / Temp 6	
348	Batt 3 / Temp 1	
349	Batt 3 / Temp 2	
350	Batt 3 / Temp 3	
351	Batt 3 / Temp 4	
352	Batt 3 / Temp 5	
353	Batt 3 / Temp 6	
354	Batt 4 / Temp 1	
355	Batt 4 / Temp 2	
356	Batt 4 / Temp 3	
357	Batt 4 / Temp 4	
358	Batt 4 / Temp 5	
359	Batt 4 / Temp 6	
360	Batt 5 / Temp 1	
361	Batt 5 / Temp 2	
362	Batt 5 / Temp 3	
363	Batt 5 / Temp 4	THERM

Appendix A cont.

Channel	Use	Voltage (Volts)
364	Batt 5 / Temp 5	THERM
365	Batt 5 / Temp 6	
366	Batt 6 / Temp 1	
367	Batt 6 / Temp 2	
368	Batt 6 / Temp 3	
369	Batt 6 / Temp 4	
370	Batt 6 / Temp 5	
371	Batt 6 / Temp 6	
372	Not Used	
373	Not Used	
374	Not Used	
375	Not Used	THERM
376	Batt 1 / VOLT	200 V
377	Batt 2 / VOLT	
378	Batt 3 / VOLT	
379	Batt 4 / VOLT	
380	Batt 5 / VOLT	
381	Batt 6 / VOLT	
382	BUS 1 / VOLT	
383	BUS 2 / VOLT	
384	BUS 3 / VOLT	
385	Not Used	
386	Not Used	
387	Not Used	
388	Not Used	
389	Not Used	
390	Not Used	
391	Not Used	
392	Not Used	
393	Not Used	
394	Not Used	
395	Not Used	
396	Not Used	
397	Not Used	
398	Not Used	
399	Not Used	200 V
400	Batt 1 / CURR	±100 MV
401	Batt 2 / CURR	
402	Batt 3 / CURR	
403	Batt 4 / CURR	
404	Batt 5 / CURR	
405	Batt 6 / CURR	
406	BPRC 1 / CURR	
407	BPRC 2 / CURR	
408	BPRC 3 / CURR	
409	BPRC 4 / CURR	
410	BPRC 5 / CURR	
411	BPRC 6 / CURR	±100 MV

Appendix B
Digital inputs

Channel	Type	Function
1	Isolated	POWER FAILURE
2	Isolated	PRT BATT 1
3	Isolated	PRT BATT 2
4	Isolated	PRT BATT 3
5	Isolated	PRT BATT 4
6	Isolated	PRT BATT 5
7	Isolated	PRT BATT 6
8	Isolated	BUS PNT
9	Isolated	TAPE OFF LINE
10	Isolated	
11	Isolated	
12	Isolated	FORCE DAY
13	Isolated	BEGIN NIGHT
14	Isolated	BEGIN DAY
15	Isolated	FORCE NIGHT
16	Isolated	Not Used
17	Nonisolated	
18	Nonisolated	RECONDITION BATT 1
19	Nonisolated	RECONDITION BATT 2
20	Nonisolated	RECONDITION BATT 3
21	Nonisolated	RECONDITION BATT 4
22	Nonisolated	RECONDITION BATT 5
23	Nonisolated	RECONDITION BATT 6
24	Nonisolated	
25	Nonisolated	
26	Nonisolated	
27	Nonisolated	1PPM
28	Nonisolated	TIME BCD 1
29	Nonisolated	TIME BCD 2
30	Nonisolated	TIME BCD 4
31	Nonisolated	TIME BCD 8
32	Nonisolated	Not Used

Appendix C
Digital outputs

Channel	Type	Function
1-0	Isolated	Not Used
1-1		DAY
1-2		NIGHT
1-3		Not Used
1-4		
1-5		
1-6		
1-7		
1-8		
1-9		
1-10		
1-11		
1-12		
1-13		
1-14		
1-15	Isolated	ALARM ON
2-0	Nonisolated	Not Used
2-1		SCANNING
2-2		SCANNING
2-3		Not Used
2-4		
2-5		
2-6		
2-7		
2-8		
2-9		
2-10		
2-11		
2-12		
2-13		
2-14		
2-15		
2-16		
2-17		
2-18		
2-19		
2-20		
2-21		
2-22		
2-23		
2-24		
2-25		
2-26		
2-27		
2-28		
2-29		
2-30		
2-31	Nonisolated	Not Used

Appendix C Cont.
Digital Outputs

Channel	Type	Function
2-32	Nonisolated	Not Used
2-33		
2-34		
2-35		
2-36		
2-37		
2-38		
2-39		
2-40		
2-41		
2-42		
2-43		
2-44		
2-45		
2-46		
2-47		
2-48		
2-49		
2-50		
2-51		
2-52		
2-53		
2-54		
2-55		
2-56		
2-57		
2-58		
2-59		
2-60		1PPM RS
2-61		TIME OFF
2-62		TIME ADV
2-63	Nonisolated	TIME HOLD

APPENDIX D

HUBBLE SPACE TELESCOPE
NICD 6 BATTERY SYSTEM
TEST OPERATIONS PROGRAM WITH COMMENTS
REVISED FEB 27, 1990

DATA ACQUISITION SYSTEM STARTUP PROCEDURE

1. DADS RACK MAIN POWER SWITCH "ON" (BOTTOM PANEL)
2. RESET EXTERNAL CLOCK (TOP PANEL) & PRESS "STOP" SWITCH
3. SET EXTERNAL CLOCK TO CURRENT DAY & TIME
4. START CLOCK
5. TURN ON VIDEO DISPLAY
   MESSAGE SHOULD READ: (AFTER BEEP)
   V14F1 TEST OK
6. TURN ON PRINTER A AND VERIFY IS "ON LINE"
7. TURN ON PRINTER B AND VERIFY IT IS "ON LINE"
8. CONNECT MODEM
9. TURN ON LOWER "CAMAC" CRATE
10. TURN ON UPPER "CAMAC" CRATE
11. TURN ON BACKPLANE POWER (SHORT RACK)
    ENABLE SWITCH "UP", LTC SWITCH "UP"
12. TURN ON MAG TAPE POWER SWITCH
13. PUT "TAPE OFF LINE" SWITCH UP (AUX PANEL)
14. INSERT DISK "A" INTO DRIVE "0" (LEFT)
15. INSERT DISK "B" INTO DRIVE "1" (RIGHT)
16. VERIFY VIDEO DISPLAY IS OK
    MESSAGE SHOULD READ:
    V14F1 TEST OK
17. DEPRESS "BOOT" SWITCH ON UPPER "CAMAC" CRATE"
18. THE FOLLOWING WILL BE DISPLAYED
    (ENTER BOLDFACED DATA,DATA SHOWN ARE EXAMPLES ONLY)
    (BOLD LETTERS ARE OPERATOR RESPONSES)
    (<> INDICATES "RETURN" KEY)
    o D 56 = 5015
    o SET TT SCOPE
    o D56 = 0

o INITIALIZE VMO:
VMO INITIALIZE; ARE YOU SURE? **Y** <>
o COPY DYO: RT11SJ.SYS VMO:
    FILES COPIED
    DYO:RT11SJ.SYS TO VMO:RT11SJ.SYS
o COPY DYO:SWAP.SYS VMO:
    FILES COPIED
    DYO:SWAP.SYS TO VMO:SWAP.SYS
o COPY DYO:TT.SYS VMO:
    FILES COPIED
    DYO:TT.SYS TO VMO:TT.SYS
o COPY DYO:DY.SYS VMO:

```
    FILES COPIED
    DY0:DY.SYS TO VM0:DY.SYS
o COPY DY0:VM.SYS VM0:
    FILES COPIED
    DY0:VM.SYS TO VM0:VM.SYS
o COPY DY0:MS.SYS VM0:
    FILES COPIED
    DY0:MS.SYS TO VM0:MS.SYS
o COPY DY0:LS.SYS VM0:
    FILES COPIED
    DY0:LS.SYS TO VM0:LS.SYS
o COPY DY0:NL.SYS VM0:
    FILES COPIED
    DY0:NL.SYS TO VM0:NL.SYS
o COPY DY0:LT.SYS VM0:
    FILES COPIED
    DY0:LT.SYS TO VM0:LT.SYS
o COPY DY0:TV.SYS VM0:
    FILES COPIED
    DY0:TV.SYS TO VM0:TV.SYS
o COPY DY0:DIR.SYS VM0:
    FILES COPIED
    DY0:DIR.SYS TO VM0:DIR.SYS
o COPY DY0:PIP.SYS VM0:
    FILES COPIED
    DY0:PIP.SYS TO VM0:PIP.SYS
o COPY DY0:DUP.SAV VM0:
    FILES COPIED
    DY0:DUP.SAV TO VM0:DUP.SAV
o COPY DY0:RESORC.SAV VM0:
    FILES COPIED
    DY0:RESORC.SAV TO VM0:RESORC.SAV
o COPY DY1:INIT.BAS VM0:
    FILES COPIED
    DY1:INIT.BAS TO VM0:INIT.BAS
o COPY DY1: MAINOP.BAS VM0:
    FILES COPIED
    DY1: MAINOP.BAS TO VM0:MAINOP.BAS
o COPY DY0:BASICO.SAV VM0:
    FILES COPIED
    DY0:BASICO.SAV TO VM0:BASICO.SAV
o COPY DY0:STARTX.COM VM0:STARTS.COM
o COPY BOOT VM0:RT11SJ.SYS VM0:
o BOOT VM0:
o RT-11SJ (S) V05-01
o D 56 = 5015
```

```
o SET LS FORM, WIDTH = 132
o ASS LS LP
o SET TT SCOPE
o SET EDIT KED
o D 56 = 0
o DATE 2-APR-86 <>
o TIME 8:17:54 <>


A.
o RUN BASICO <>
BASIC-11/RT-11 V02-03
OPTIONAL FUNCTIONS (ALL, NONE, OR INDIVIDUAL)? A <>
READY
RUN INIT <>
CRATE 4 STATUS
E205 INSTALLED IN STATION 1
E220 INSTALLED IN STATION 2
E220 INSTALLED IN STATION 3
E220 INSTALLED IN STATION 4
E220 INSTALLED IN STATION 5
E220 INSTALLED IN STATION 6
E220 INSTALLED IN STATION 7
E220 INSTALLED IN STATION 8
E220 INSTALLED IN STATION 9
E220 INSTALLED IN STATION 10
E220 INSTALLED IN STATION 11
E220 INSTALLED IN STATION 12
E220 INSTALLED IN STATION 13
E220 INSTALLED IN STATION 14
E220 INSTALLED IN STATION 15
E220 INSTALLED IN STATION 16
E220 INSTALLED IN STATION 17
ENTER 1 TO CONTINUE
? 1 <>
CRATE 5 STATUS
E120 INSTALLED IN STATION 2
E140 INSTALLED IN STATION 3
```

```
E140 INSTALLED IN STATION 4
E140 INSTALLED IN STATION 5
```

```
E205 INSTALLED IN STATION 6
E240 INSTALLED IN STATION 7
E205 INSTALLED IN STATION 8
E220 INSTALLED IN STATION 9
E220 INSTALLED IN STATION 10
E220 INSTALLED IN STATION 11
E220 INSTALLED IN STATION 12
E220 INSTALLED IN STATION 13
E220 INSTALLED IN STATION 14
E220 INSTALLED IN STATION 15
E220 INSTALLED IN STATION 16
E220 INSTALLED IN STATION 17
ENTER 1 TO CONTINUE
? 1 <>
ENTER ORBIT COUNT
? 1 <>
ENTER CURRENT YEAR
? 1986 <>
ENTER DISCHARGE MIN.
? 35 <>
ENTER CHARGE MIN.
? 60 <>
ENTER ORBITS TO SKIP
? 0 <>
(DELAY TO NEXT LINE WILL BE APPROX 10 SEC.)
STOP AT LINE 590
(ANY SET UP CHANGES MAY BE MADE AT THIS POINT)
(DISCH MIN = C3 = )
(CHARGE MIN = C1 = )
(SHUTDOWN FLAG = F9 = 0)
(BATT'S 1,2,3 HI LIMIT = L1= )
(BATT'S 4,5,6 HI LIMIT = L2= )
(ALL BATTS LO LIMIT = L8=     )
(ORBIT NO. = O1 = )
(ORBITS TO SKIP = S1 = )
```

```
GOTO 600 <>
(PROGRAM & SCANNING START WITH NIGHT [DISCHARGE] AND SCAN OF
0)
(THE FOLLOWING WILL BE DISPLAYED AFTER COMPLETION OF EACH
SCAN)
"DISCHARGE TIME 1 C3=35 ORBIT 1" OR "CHARGE TIME 1 C1=60
ORBIT 1"
AFTER CHECKING FOR TIME DISPLAY ABOVE, MAGNETIC TAPE MAY BE
PUT ON LINE BY PUTTING "TAPE OFF LINE" SWITCH TO THE DOWN
POSITION.
```

VARIABLE DESCRIPTIONS

```
A         =Average temperature in hi/lo subroutine.
A1(6)     =Amp-min in summation.
A2(6)     =Amp-min out summation.
A3(6)     =BPRC ahr summation.
A6(6)     =Orbit high pressure.
A7(6)     =Orbit high pressure cell no.
A8(3)     =Bus current.
A9(3)     =Bus voltage.

B1(6)     =Discharge hi cell voltage.
B2(6)     =Discharge hi cell no.
B3(6)     =Discharge lo cell voltage.
B4(6)     =Discharge lo cell no.
B5(6)     =Discharge average cell voltage.
B6(6)     =Discharge battery voltage at lo cell.
B7(7)     =Discharge hi temperature.
B8(6)     =Discharge lo temperature.
B9(6)     =Discharge average temperature.

C1        =Charge time of orbit.
C2        =Charge elapsed min. counter.
C3        =Discharge time of orbit.
C4        =discharge elapsed min. counter.
C5(6)     =Reconditioning counter.
C8        =Print value of C2.
C9        =Print value of C4.
C$        =":"

D1(6,23)=Cell voltage.
D2(6)     =Battery voltage.
D3(6)     =Battery current.
D5(6)     =Battery temperature.
D6(13)    =SAS/PS current.
D7(6)     =Last scan current.
D8(6,23)=Cell pressure.
D9(6)     =BPRC current.

E1(6)     =Charge hi cell voltage.
E2(6)     =Charge hi cell no.
E3(6)     =Charge lo cell voltage.
E4(6)     =Charge lo cell no.
E5(6)     =Charge average cell voltage.
E6(6)     =Charge battery volts at hi cell.
E7(6)     =Charge hi temperature.
E8(6)     =Charge lo temperature.
E9(6)     =Charge average temperature.

F1(0)     =Dig. input, power fail.
```

```
F1(1)    =Dig. input, battery 1 print request.
F1(2)    =Dig. input, battery 2 print request.
F1(3)    =Dig. input, battery 3 print request.
F1(4)    =Dig. input, battery 4 print request.
F1(5)    =Dig. input, battery 5 print request.
F1(6)    =Dig. input, battery 6 print request.
F1(7)    =Dig. input, bus data print request.
F1(8)    =Dig. input, tape off line.
F1(9)    =Dig. input, skip scan on mag. tape.
F1(10)   =
F1(11)   =Dig. input, force day.
F1(12)   =Dig. input, begin night.
F1(13)   =Dig. input, begin day.
F1(14)   =Dig, input, force night.
F1(15)   =bad
F2(0)    =
F2(1)    =Dig. input, reconditioning battery 1.
F2(2)    =Dig. input, reconditioning battery 2.
F2(3)    =Dig. input, reconditioning battery 3.
F2(4)    =Dig. input, reconditioning battery 4.
F2(5)    =Dig. input, reconditioning battery 5.
F2(6)    =Dig. input, reconditioning battery 6.
F2(7)    =Dig. input, trickle charge.
F2(8)    =
F2(9)    =
F2(10)   =Dig. input, 1ppm log pulse.
F2(11)   =Dig. input, time BCD 1.
F2(12)   =Dig. input, time BCD 2.
F2(13)   =Dig. input, time BCD 4.
F2(14)   =Dig. input, time BCD 8.
F2(15)   =bad
F3       =Roll flag.
F7       =SOC reset flag.
F8       =Descrete trickle flag.
F9       =Shutdown flag.

G1       =Temporary battery WHO for print.
G2       =Temporary battery EFF for print.
G3       =Temporary battery RR for print.
G4       =Temporary battery DCH or Dcc for print.
G5       =Temoprary battery DOD for print.
G6       =Temporary battery AHO for print.
G7       =Temporary battery BPRC_WH for print.

H        =Hi temperature in hi\lo temperature routine.

I        =Temporary counter, usually battery no.
I1       =Temporary counter.

J        =Temporary counter, usually cell no.
J1       =Temporary counter.

K1       =AHO constant (1/60).
```

```
L         =Lo temperature in hi\lo temperature routine.
L1        =Battery 1,2,3 hi cell limit.
L2        =Battery 4,5,6 hi cell limit.
L8        =Battery lo limit.
L9        =Current cell limit in use.

M%        =Multiplexor and ADC setup values.
M1        =Mag tape off line flag.

N         =Temporary no. storage.
N1(9)     =Temporary no. storage in time calc..

O1        =Orbit no.

P1        =Battery "printed" flag.
P2        =Phase flag, 0= disc.,1=chrg.
P3        =Forced night flag.
P4        =Forced day flag.
P5        =Printout phase flag
P8        =Flag to check P.S. during 1st 5 min of day.
P9        =Flag to check load during night.

R         =Temporary resistor value.
R2        =Constant for DOD calc. (1.01)
R9(5,9)   =RR from 10 orbits.

S1        =Orbits to skip value.
S2        =Skip counter.
S3        =Skip scan counter.
S4(6)     =SOC max value.
S5(6)     =SOC accumulative.
S6        =SOC print value.
S7(6)     =SOD print value.

T1(4)     =Storage for scan time.
T2(4)     =Converted time characters.
T3(6,4)   =Sunset time.
T9        =No. of scans on tape counter.

U1        =Print request counter.
U2        =Retry counter.

V         =Temporary value in temperature calc.
V1        =Power fail flag.
V2        =First scan flag.
V3(6)     =Reconditioning flag.
V4(6)     =Capacity test complete flag.
V5(13)    =SAS/PS failure flag.
V6(6)     =Battery current failure flag.
V7        =Begin night flag.
V8        =Begin day flag.

W1(6)     =Battery watt-min in summation.
W2(6)     =Battery watt-min out summation.
```

```
Z1%       =ADC and switch reg. output values.
Z2%       =Temporary output value.
```

WHERE TO CHANGE NON-VARIABLE LIMITS

LINE NO.	DESCRIPTION	CURRENT VALUE
565	SOC BASE VALUE	55
2700	BATT RECOND. 1ST LIMIT	26.45 (VOLTS)
3020	BATT. I UPPER LIMIT AT NIGHT	5 (AMPS)
4073	EXTRANEOUS CELL VOLTAGE DEVIATION BETWEEN SCANS (EXCEPT FIRST OF PHASE)	.01 (VOLTS)
4151	EXTRANEOUS CELL PRESS. DEVIATION	10 (PSI)
4241	EXTRANEOUS BATT. V DEV.	.1 (VOLTS)
4311	CORRECTION FACTORS FOR CURRENTS	
4390-4420	BATT TEMP. O.C. VALUE	2.2303 (VOLTS)
4490	BATT. TEMP. SHRT. CIR.	-1.2575 (VOLTS)
4500	BATT. DISC. CURR. LIMIT	-30 (AMPS)
7080	BATT. CHRG. CURR. LIMIT	25 (AMPS)
7090	SAS P.S. DISC. LIMIT	5 (AMPS)
7620	SAS P.S. CHRG 5 MIN LIMIT	5 (AMPS)
7830	BATT. AVE. HI TEMP. LIMIT	30 (DEG. C)
8580	BATT. AVE. LO TEMP. LIMIT	-10 (DEG. C)
8590		

```
THIS SECTION IS THE FIRST PROGRAM GROUP AND WILL
CHAIN TO THE MAIN OPERATING PROGRAM WHEN COMPLETE

1     REM "INIT" Routine to initialize crates and modules
      *Define the variables common to both program sections
50    COMMON O1,C3,C1,T1(4),S1
      *The next section initializes crate #4 and checks for
      *the module codes.
100   CALL CRATE(4) \ CALL INIT \ CALL INH(0)
      *Print status header
110   PRINT "CRATE 4 STATUS"
      *Set up module loop
120   FOR J=1 TO 17
      *Get module ID numbers
130   CALL C(J,0,1,I)
      *Print module ID no.
140   PRINT "E";I;" INSTALLED IN POSITION ";J
      Get another module ID
150   NEXT J
      *Print operator instruction
160   PRINT "ENTER 1 TO CONTINUE"
      *Get input
170   INPUT J1
      *Verify input is a 1
180   IF J1<>1 GOTO 160
      *The next part initializes crate #5 and checks
      *the ID of all modules
190   CALL CRATE(5) \ CALL INIT \ CALL INH(0)
      *Print status header
200   PRINT "CRATE 5 STATUS"
      *Set up module loop
210   FOR J=2 TO 17
      *Get module ID's
220   CALL C(J,0,1,I)
      *Print status
230   PRINT "E ";I;" INSTALLED IN POSITION ";J
      *Next module
240   NEXT J
      *Print instruction
250   PRINT "ENTER 1 TO CONTINUE"
      *Get input
260   INPUT J1
      *Verify input is a 1
270   IF J1<>1 GOTO 250
      *This section gets the starting orbit no. and year
      *Print instruction
280   PRINT "ENTER ORBIT COUNT"
      *Get starting orbit count
290   INPUT O1
      *Print instruction
300   PRINT "ENTER CURRENT YEAR"
      *Get current year
```

```
310  INPUT T1(0)          **T1(0) IS YEAR IN TIME ARRAY
     *print instruction
320  PRINT "ENTER DISCHARGE MIN."
     *Get discharge time
330  INPUT C3             **C3 IS LENGTH OF NIGHT
     *print instruction
340  PRINT "ENTER CHARGE MIN."
     *Get charge time
350  INPUT C1             **C1 IS LENGTH OF DAY
     *Print instruction
360  PRINT "ENTER ORBITS TO SKIP"
     *Get skip count
370  INPUT S1
     *clear clock buffers and send clock hold pulse
400  CALL C(4,1,16,(2^15)) \ CALL C(4,1,16,0)
     *Set up time input loop
410  FOR I=1 TO 9
     *Set up to get digit
420  CALL C(4,1,16,(2^14)) \ CALL C(4,1,16,0)
     *Get a time digit
430  CALL C(2,1,0,Z1%)
     *Get another
440  NEXT I
     *Reset counter and reset hold
450 CALL C(4,1,16,(2^15)) \ CALL C(4,1,16,(2^13)) \ CALL
    C(4,1,16,0)
     *Go to next program segment
500  CHAIN "MAINOP"
510  END
```

```
THIS STARTS THE MAIN OPERATING SECTION OF THE PROGRAM

2     REM "MAINOP" main operating program
      *Set up common variables
50    COMMON O1,C1,C3,T1(4),S1
      *Dimension arrays
500   DIM A1(6),A2(6),B1(6),B2(6),B3(6),B4(6),B5(6)
510   DIM B6(6),B7(6),B8(6),B9(6),C5(6),D2(6),D3(6)
520   DIM D6(13),D7(6),E1(6),E2(6),E3(6),E4(6),E5(6)
530   DIM E6(6),E7(6),E8(6),E9(6),F1(16),F2(16)
540   DIM T2(9),V3(7),V4(6),V5(13),V6(6),W1(6),W2(6)
550   DIM D9(6),A8(3),A9(3),A6(6),A7(6),A4(6)
555   DIM S4(6),S5(6),S6(6),S7(6)
560   DIM D1(6,23),D5(6,6),D8(6,23),T3(6,4),R9(5,9)
565   S6=48 \ FOR I=1 TO 6 \ S4(I)=S6 \ NEXT I
      *The next section initializes flags and summations
570   FOR I=1 TO 6 \ B3(I)=2 \ S5(I)=S4(I) \ NEXT I
575   FOR I=0 TO 5 \ FOR J=0 TO 9 \ R9(I,J)=0 \ NEXT J \ NEXT
I
580   L1=1.55 \ L2=1.55 \ L8=0 \ C$=":" \ C2=0 \ C4=0 \ P2=0 \
V1=0 \ V2=0
      *stop here to catch up ( type continue to resume)
590   STOP
      *Set crate no. and set to discharge phase
600   CALL CRATE(5) \ CALL C(5,0,16,2^2) \ CALL C(5,0,16,0)
      *Clear 1ppm log input
610   CALL C(4,1,16,(2^12)) \ CALL C(4,1,16,0)
      *Set up initial values
620   S2=0 \ M1=1 \ P8=0 \ P9=0 \ J9=0
630   R2=1.01 \ F3=0 \ K1=1/60
640   T8=0
      *Get input switch closures
1000  GOSUB 5000
      *check for shutdown
1010  IF F9=1 GOTO 1900
      *check power failure signal if zero skip to next check
1020  IF F1(0)=1 GOTO 1080
      *check power fail flag, if set go check signal again
1030  IF V1=1 GOTO 1900
      *power fail signal "on" flag not set so print out notice
       header
1040  GOSUB 1810
      *print power fail message
1050  PRINT #2,"POWER FAILED SINCE"
      *print blanks, close output, & set power fail flag
1060  PRINT #2 \ CLOSE #2 \ V1=1
      *go beginning to check for power return
1070  GOTO 1000
      *if flag is zero continue normal input check
1080  IF V1=0 GOTO 1150
      *power was off, now on, get current time
1090  CALL C(4,1,16,(2^15)) \ CALL C(4,1,16,0) \ GOSUB 5200
```

```
     *open output line & print header
1100 GOSUB 1810
     *print out message
1110 PRINT #2,"POWER RETURNED"
     *close output line & reset flag
1120 PRINT #2 \ CLOSE #2 \ V1=0
     *clear out 1ppm signal
1130 CALL C(4,1,16,(2^12)) \ CALL C(4,1,16,0)
     *go back to checking input signals
1140 GOTO 1000
     *if flag set go scan branch
1150 IF F2(10) = 1 GOTO 2000
     *set up battery loop to check print requests
1160 FOR I=1 TO 7
     *if recond sw set, set recond flag
1170 IF F2(I)=1 THEN V3(I)=1
     *if printed flag set skip printing
1180 IF P1=1 GOTO 1210
     *if print flag not set go check another
1190 IF F1(I)=0 GOTO 1210
     *if print requested set flag
1200 U1=1
     *get next
1210 NEXT I
     *if prints not requested skip
1220 IF U1=0 GOTO 1400
     *print header
1230 GOSUB 1810
     *set up battery loop
1240 FOR I=1 TO 6
     *skip if battery print not requested
1250 IF F1(I)=0 GOTO 1340
     *print battery values
1270 PRINT #2,"BATTERY=";I;"V=";D2(I);"I=";
     D3(I);"BPRC I=";D9(I);"SOC=";S5(I);"AHO=";(-1*A2(I))/60
     *set up to print cell voltages
1280 A=1 \ N=12 \ GOSUB 1850
     *print some more
1290 PRINT #2 \ A=13 \ N=23 \ GOSUB 1850
     *print temperatures
1300 PRINT #2 \ FOR I1=1 TO 6 \ PRINT #2,USING "
     ###.##",D5(I,I1) ; \ NEXT I1 \ PRINT #2
1310 FOR I=1 TO 12 \ PRINT #2, USING " ###.## ",D8(I,I1);
\ NEXT I1 \ PRINT #2
1320 FOR I(1)=13 TO 23 \ PRINT #2,USING " ###.## ", D8(I,I1);
     \ NEXT I1
     *add spaces
1330 PRINT #2 \ PRINT #2
     *get another
1340 NEXT I
     *check bus data request
1350 IF F1(7)=0 GOTO 1380
     *print header
1360 PRINT #2,"BUS    VOLTS    CURR"
```

```
     *print data
1370 FOR I=1 TO 3 \ PRINT #2,USING " #    ##.###    ##.###",
     I,A9(I),A8(I) \ NEXT I
     *close up print & set "printed" flag
1380 P1=1 \ CLOSE #2 \ U1=0
     *if sw not set skip force night routine
1400 IF F1(14) = 0 GOTO 1430
     *if already forced to night skip
1410 IF P3=1 GOTO 1430
     *force night, set flag & clear begin flags
1420 CALL C(5,0,16,2^2) \ CALL C(5,0,16,0) \ PRINT "FORCE
     NIGHT" \ P3=1 \ P4=0 \ V7=0 \ V8=0
     *if sw not set skip force day routine
1430 IF F1(11)=0 GOTO 1460
     *if already forced to day skip
1440 IF P4=1 GOTO 1460
     *force day, set flags and clear begin flags
1450 CALL C(5,0,16,2^1) \ CALL C(5,0,16,0) \ PRINT "FORCE
     DAY" \ P4=1 \ P3=0 \ V7=0 \ V8=0
     *if sw not set skip begin night routine
1460 IF F1(12)=0 GOTO 1500
     *if already begin night skip
1470 IF V7=1 GOTO 1500
     *print message & set/reset flags
1480 PRINT "BEGIN NIGHT" \ V7=1 \ P3=0 \ P4=0 \ V8=0 \ C4=0
     *go to 'end of orbit' routine
1490 GOTO 2530
     *if sw not set, skip begin day routine
1500 IF F1(13)=0 GOTO 1540
     *if already begin day skip
1510 IF V8=1 GOTO 1540
     *set day output & set phase flag to change
1520 CALL C(5,0,16,2^1) \ CALL C(5,0,16,0) \ P2=1
     *print message & set/reset flags
1530 PRINT "BEGIN DAY" \ V8=1 \ P3=0 \ P4=0 \ V7=0 \ C2=0
     *test to see if tape on line
1540 IF F1(8)=1 GOTO 1570
     *switch off was it on?
1550 IF M1=1 GOTO 1590
     *was off return to beginning
1560 GOTO 1000
     *tape off line sw. on,was it before?
1570 IF M1=1 GOTO 1000
     *new off line request, close tape
1580 CLOSE #9 \ MI=1 \ GOTO 1000
     *new on-line request, turn on tape output
1590 OPEN "MSO:DATA" FOR OUTPUT AS FILE #9, FILESIZE 1
     *set flags and return
1600 M1=0 \ T9=0 \ GOTO 1000
     *send out shutdown closure & set flag
1790 CALL C(5,0,16,(2^14)) \ F9=1
     *send out alarm closure
1800 CALL C(5,0,16,(2^15)) \ CALL C(5,0,16,0)
     *open printer file
```

```
1810 OPEN "LT:" FOR OUTPUT AS FILE #2
     *print header data
1820 PRINT #2,T1(0);C$;T1(1);C$;T1(2);C$;T1(3);C$;T1(4);
     "ORB=";02;"P=";P5;"C2=";C8;"C4=";C9 \ RETURN
     *print cell data
1850 FOR I1=A TO N \ PRINT #2,USING " ##.#### ",D1(I,I1); \
     NEXT I1 \ RETURN
     *send ippm pulse
1900 CALL C(3,0,16,3) \ CALL C(3,0,16,0)
     *return to beginning
1910 GOTO 1000


2000 REM SCAN BRANCH
     *send "latch" pulse to clock
2010 CALL C(4,1,16,(2^12+2^15)) \ CALL C(4,1,16,0)
     *send "scanning" pulses out
2020 CALL C(3,0,16,3) \ CALL C(3,0,16,0)
     *go scan all data & convert as necessary
2030 GOSUB 4000
     *go get "now" time
2040 GOSUB 5200
     *go check hi-lo limits
2050 GOSUB 7000
     *test for "initial" flag
2060 IF V2=1 GOTO 2090
     *store starting time
2070 FOR I=1 TO 6 \ FOR J=0 TO 4 \ T3(I,J)=T1(J) \ NEXT J \
     NEXT I
     *set "initial" flag
2080 V2=1
     *set up loop
2090 FOR I=1 TO 6
     *check phase
2100 IF P2=1 GOTO 2130
     *check for first discharge phase scan if no skip
2110 IF C4>0 GOTO 2150
     *skip
2120 GOTO 2140
     *check for first phase scan if no skip
2130 IF C2>0 GOTO 2150
     *set up last scan value
2140 D7(I)=D3(I)
     *calculate current average
2150 N=(D3(I)+D7(I))/2
2152 IF D3(I)>0 GOTO 2154
2153 S5(I)=S5(I)+N*K1 \ GOTO 2155
2154 S5(I)=S5(I)+N*K1/R2
2155 IF F3=0 GOTO 2160
2156 IF P2=1 GOTO 2170
     *check for negative current
2160 IF D3(I)<0 GOTO 2180
    *calculate amps. in
2170 A1(I)=A1(I)+N \ W1(I)=W1(I)+(N*D2(I)) \ GOTO 2190
```

```
     *calculate amps. out
2180 A2(I)=A2(I)+N \ W2(I)=W2(I)+(N*D2(I))
     *update last current
2190 D7(I)=D3(I) \ (A3(I)=A3(I)+D9(I))
     *temporarily store phase flag
2191 Y1=P2
     *check trickle charge flag
2194 IF F8=0 GOTO 2200
     *tickles charge on, reset SOC values
2196 S5(I)=S4(I) \ F7=1 \ T8=C2
     *set trickle charge flag for output
2197 Y1=P2+10
2200 IF P2=0 GOTO 2230
     *in charge find hi cells
2210 GOSUB 8350
     *check if still in recond.
2230 IF V3(I)=1 GOTO 2670
     *get another battery
2240 NEXT I
2245 IF F7=1 THEN F8=0
     *check load bank total current
2250 A8(0)=A8(1)+A8(2)+A8(3) \ IF A8(0)<99 GOTO 2280
     *out of limit go to shutdown
2260 GOSUB 1790
     *print message
2270 PRINT #2,"TOTAL BUS I=";A8(0) \ CLOSE #2
     *transmit data to smart system & mag tape
2280 GOSUB 5500
     *set print parameters
2290 P5=P2 \ 02=01
     *check phase
2300 IF P2=1 GOTO 2430
     *update discharge counter & display
2310 C4=C4+1 \ PRINT "DISC MIN=";C4;"C3=";C3;"ORBIT NO.";
     O1 \ C9=C4 \ C8=C2
2320 GOSUB 3000
     *check for "Forced Day"
2330 IF P4=1 GOTO 2350
     *go check power supply limits
2340 GOSUB 7600
     *discharge time not up return to main
2350 IF C4<C3 GOTO 1000
     *set up loop
2360 FOR I=1 TO 6
2362 S7(I)=S5(I)
     *go check disch. hi-lo cells
2370 GOSUB 8200
     *go get E-O-D H-L temps
2380 GOSUB 8500
     *store results
2390 B9(I)=A \ B8(I)=L \ B7(I)=H
     *next batt.
2400 NEXT I
     *send out day pulse & set day flag * reset force & begin
```

```
     flags
2410 CALL C(5,0,16,2^1) \ CALL C(5,0,16,0) \ P2=1 \ P4=0 \
V7=0 \V8=0
     *return to main program
2420 GOTO 1000
     *update charge counter
2430 C2=C2+1 \ PRINT "CHRG MIN=";C2;"C1=";C1;"ORBIT NO.";01
     \ C8=C2
     *go check power supply limits
2440 GOSUB 7800
     *if charge time not up go to main
2450 IF C2<C1 GOTO 1000
     *update orbit count
2460 O1=O1+1
     *update orbit skip counter
2470 S2=S2+1
     *check its value
2480 IF S2>0 THEN S2=-S1
     *set up loop
2490 FOR I=1 TO 6
     *go get END-O-PHASE temps
2496 S6(I)=S5(I)
2500 GOSUB 8500
     *store results
2510 E9(I)=A \ E8(I)=L \ E7(I)=H
     *go get next
2520 NEXT I
     *send out night pulse & set day flags & reset force -
     begin flags
2530 CALL C(5,0,16,2^2) \ CALL C(5,0,16,0) \ P2=0 \ P3=0 \
     V7=0 \ V8=0
     *set up loop
2540 OPEN "LS:" FOR OUTPUT AS FILE #1
     *open line to printer
2550 FOR I=1 TO 6
     *if in recon. skip printer
2560 IF V3(I)=1 GOTO 2590
     *go print batt results
2570 G4=C4
     *go print batt. orbital data
2580 GOSUB 6000
     *next batt
2590 NEXT I
     *close print
2595 T8=99
2600 CLOSE #1
     *zero chrg-disch counters
2603 J9=J9+1
2604 IF J9>9 THEN J9=0
2610 C2=0 \ C4=0
     *set up batt loop
2620 FOR I=1 TO 6
     *test for recond. skip time if yes
2630 IF V3(I)=1 GOTO 2650
```

```
     *load new sunset time
2640 FOR J=0 TO 4 \ T3(I,J)=T1(J) \ NEXT J
     *next batt.
2650 NEXT I
     *go to main
2660 GOTO 1000
     *update discharge recond counter
2670 C5(I)=C5(I)+1
     *check if recond still on
2680 IF F2(I)=0 GOTO 2770
     *check if captest complete
2690 IF V4(I)=1 GOTO 2240
     *check for end of captest
2700 IF D2(I)>26.45 GOTO 2240
     *set captest complete flag
2710 V4(I) = 1
     *set disch min, to captest time
2720 G4=C5(I)
     *open output for captest info
2730 OPEN "LS:" FOR OUTPUT AS FILE #1
2735 GOSUB 8200
     *go print orb (captest) info
2740 GOSUB 6000
     *close output
2750 CLOSE #1
     *return to next batt.
2760 GOTO 2240
     *recond over clear flags & set up output
2770 V3(I)=0 \ V4(I)=0 \ G4=C5(I)
     *clear disch (captest) counter
2780 C5(I) = 0
     *go print summary data
2790 GOTO 2730
     *subroutine to check bus current for day
3000 IF P9=0 GOTO 3060
     *set up loop
3010 FOR I=1 TO 3
     *check for good bus-current
3020 IF A8(I)>5 GOTO 3050
     *bad limit go alarm
3030 GOSUB 1800
     *print message
3040 PRINT #2, "BUS No.";I;"CURR=";A8(I) \ CLOSE #2
     *get another
3050 NEXT I
     *return
3060 RETURN
4000 REM SCAN DATA
     *set up crate & clear begin/force flags
4010 CALL CRATE (4)
     *set up battery loop
4020 FOR I=1 TO 6
     *set up cell loop
4030 FOR J=1 TO 23
```

```
     *set up char. no.
4040 M%=23*(I-1)+J+512 \ H=D1(I,J) \ U2=0
     *set up mux & adc at gain of 4 (+-2.56v)
4050 CALL C(1,0,16,M%)
     *set LAM status
4051 CALL C(1,0,27,,Q,X)
     *if not set try again
4052 IF Q=0 GOTO 4051
     *get adc results
4060 CALL C(1,1,0,Z1%)
     *convert to voltage
4070 D1(I,J)=(Z1%*78.125*10^(-6))
     *check for initial phase scan
4071 IF C4=0 GOTO 4080
4072 IF C2=0 GOTO 4080
4073 IF ABS(H-D1(I,J))<.01 GOTO 4080
     *check tries
4074 IF U2>2 GOTO 4080
     *incrmnt counter & go try again
4075 U2=U2+1 \ GOTO 4050
     *next cell
4080 NEXT J
     *next batt.
4090 NEXT I
     *set up batt. loop
4100 FOR I=1 TO 5
     *set up cell loop
4110 FOR J=1 TO 23
     *set up char no.
4120 M%=652+23*(I-1)+J \ H=D8(I,J) \ U2=0
     *set up mux & adc (gain 1 +-2.56v)
4130 CALL C(1,0,16,M%)
     *check if conversion complete
4131 CALL C(1,0,27,,Q,X)
     *if not ready try again
4132 IF Q=0 GOTO 4131
     *get results
4140 CALL C(1,1,0,Z1%)
     *convert to voltage & pressure (30mv=150psig)
4150 D8(I,J)=(Z1%*78.125*10^(-5))*4.97
     *check jump in reading
4151 IF ABS (H-D8(I,J))<10 GOTO 4160
     *check if enough tries
4152 IF U2>2 GOTO 4160
     *not too many try again
4153 U2=U2+1 \ GOTO 4130
     *check for highest pressure
4160 IF A6(I)<D8(I,J) THEN A6(I)=D8(I,J) \ A7(I)=J
     *next cell
4170 NEXT J
     *next batt.
4180 NEXT I
     *change to crate 5
4190 CALL CRATE(5)
```

```
     *set up cell loop
4200 FOR J=1 TO 23
     *set up to check for bad readings
4210 H=D8(6,J) \ U2=0
     *set up mux & adc
4220 CALL C(8,0,16,(511+J))
     *check if conversion complete
4221 CALL C(8,0,27,,Q,X)
     *if not try again
4222 IF Q=0 GOTO 4221
     *get adc results
4230 CALL C(8,1,0,Z1%)
     *convert to volt & pressure
4240 D8(6,J)=(Z1%*78.125*10^(-5))*4.97
     *check jump in readings
4241 IF ABS(H-D8(6,J))<10 GOTO 4250
     *too many tries
4242 IF U2>2 GOTO 4250
     *go try again
4243 U2=U2+1 \ GOTO 4220
     *check for high pressure
4250 IF A6(6)<D8(6,J) THEN A6(6)=D8(6,J) \ A7(6) = J
     *next cell
4260 NEXT J
     *set up batt loop
4270 FOR I=1 TO 6
     *set up char no. & reset print flag
4280 M%=119+I \ H=D2(I) \ U2=0
     *set up mux & adc
4290 CALL C(8,0,16,M%)
     *check LAM
4291 CALL C(8,0,27,,Q,X)
4292 IF Q=0 GOTO 4291
     *get adc results
4300 CALL C(8,1,0,Z1%)
     *convert to volt
4310 D2(I)=(Z1%*6.26)/1000
     *check batt v jump
4311 IF ABS(H-D2(I))<.1 GOTO 4320
     *check tries
4312 IF U2>2 GOTO 4320
     *try again
4313 U2=U2+1 \ GOTO 4290
     *set up mux * adc
4320 CALL C(6,0,16,(I-1))
     *check LAM
4321 CALL C(6,0,27,,Q,X)
     *if not ready try again
4322 IF Q=0 GOTO 4321
     *get results
4330 CALL C(6,1,0,Z1%)
     *convert to volt & store it batt I
4340 D3(I)=Z1%*2.50000E-03
     *set up mux for BPRC current
```

```
4350 CALL C(6,0,16,(517+I))
     *check LAM
4351 CALL C(6,0,27,,Q,X)
     *if not ready try again
4352 IF Q=0 GOTO 4351
     *get results
4360 CALL C(6,1,0,Z1%)
     *convert and store BPRC current
4370 D9(I)=Z1%*6.25000E-04
     *next battery
4380 NEXT I
     *current correction factors
4390 D3(1)=D3(1)+0\D3(2)=D3(2)+0\D3(3)=D3(3)+0
4400 D3(4)=D3(4)-0\D3(5)=D3(5)+0\D3(6)=D3(6)+0
4410 D9(1)=D9(1)-2.00000E-03 \ D9(2)=D9(2)-1.00000E-03 \
D9(3)=D9(3)-8.00000E-03
4420 D9(4)=D9(4)-7.00000E-03 \ D9(5)=D9(5)-6.00000E-03 \
D9(6)=D9(6)-.011
     *set up batt loop
4430 FOR I=1 TO 6
     *set up meas. loop
4440 FOR J=1 TO 6
     *set up char no. & adc data
4450 M%=335+6*(I-1)+J
     *set up adc & mux
4460 CALL C(8,0,16,M%)
     *get results
4470 CALL C(8,1,0,Z1%)
     *convert to volts
4480 V=Z1%*.15625/1000
     *test for too high
4490 IF V>2.2303 THEN V=2.2303
     *test for too low
4500 IF V<-1.2575 THEN V=-1.2575
     *convert to res.
4510 R=((V+2.515)/(2.515-V))*3000
     *convert to deg c
4520 D5(I,J)=(5153.2/(LOG(R)+7.03721))-317.688
     *next meas
4530 NEXT J
     *next batt.
4540 NEXT I
4542 D5(2,3)=(D5(2,1)+D5(2,2)+D5(2,4)+D5(2,5)+D5(2,6))/5
     *set up loop for sas/ps.
4550 FOR I=1 TO 13
     *set up mux & adc data
4560 M%=578+I
     *set up mux & adc
4570 CALL C(8,0,16,M%)
     *check to see if conversion complete
4571 CALL C(8,0,27,,Q,X)
     *if not done try again
4572 IF Q=0 GOTO 4571
     *get results
```

```
4580 CALL C(8,1,0,Z1%)
     *convert to amps
4590 D6(I)=Z1%*78.125*10^(-3)
     *next sas
4600 NEXT I
     *set up loop for bus data
4610 FOR I=1 TO 3
     *adjust range and chan
4620 M%=575+I
     *set up mux
4630 CALL C(8,0,16,M%)
     *check to see if conversion complete
4631 CALL C(8,0,27,,Q,X)
     *if not done try again
4632 IF Q=0 GOTO 4631
     *get result
4640 CALL C(8,1,0,Z1%)
     *convert data
4650 A8(I)=(Z1%*78.125*10^(-5)*2)
     *adjust chan
4660 M%=125+I
     *set up mux
4670 CALL C(8,0,16,M%)
     *get results
4680 CALL C(8,1,0,Z1%)
     *convert to volts
4690 A9(I)=(Z1%*6.26)/1000
     *next bus
4700 NEXT I
     *reset print flag
4710 P1=0
4720 RETURN
5000 REM SUBROUTINE TO GET SWITCH CLOSURES
     *set up crate/get ok from module
5010 CALL CRATE(5)
     *get first set of inputs (iso-inputs)
5020 CALL C(2,0,0,Z1%)
     *get second set of inputs (non-iso)
5030 CALL C(2,1,0,Z2%)
     *convert both words to proper format
5040 Z1%=ABS(1+Z1%) \ Z2%=ABS(1+Z2%)
     *set up loop to check inputs
5050 FOR L=0 TO 15
     *set up compare valve & clear flags
5060 N=2^(15-L) \ F1(15-L)=0 \ F2(15-L)=0
     *compare a flag if no-go skip to next check
5070 IF Z1%-N<0 GOTO 5090
     *reduce input by compared valve & set flag
5080 Z1%=Z1%-N \ F1(15-L)=1
     *compare a flag if no-go skip to next check
5090 IF Z2%-N<0 GOTO 5110
     *reduce input by compared valve & set flag
5100 Z2%=Z2%-N \ F2(15-L)=1
     *get next get to compare
```

```
5110 NEXT L
     *return to main program
5120 IF F2(7)=0 GOTO 5150
5130 IF F7=1 GOTO 5160
5140 F8=1 \ GOTO 5160
5150 F7=0
5160 RETURN
5200 REM GETIME
     *set up crate & set hold/reset
5210 CALL CRATE(5)
     *set up loop
5220 FOR I=1 TO 9
     *advance char. counter in clock
5230 CALL C(4,1,16,(2^14)) \ CALL C(4,1,16,0)
     *get inputs
5240 CALL C(2,1,0,Z1%)
     *adjust results
5250 N1(I)=ABS(1+Z1%)
     *next input
5260 NEXT I
     *reset counter \ and release hold
5270 CALL C(4,1,16,(2^15)) \ CALL C(4,1,16,(2^13)) \ CALL C(
     4,1,16,0)
     *set up decode loop
5280 FOR I=1 TO 9
     *clear clock char
5290 T2(I)=0
     *set up bit loop
5300 FOR J=0 TO 3
     *set up compare no.
5310 N=2^(14-J)
     *compare bits if yes continue
5320 IF N1(I)-N<0 GOTO 5350
     *adjust value
5330 N1(I)=N1(I)-N
     *add bit to char word
5340 T2(I)=T2(I)+2^(3-J)
     *next bit
5350 NEXT J
     *next char
5360 NEXT I
     *build days
5370 T1(1)=100*T2(1)+10*T2(2)+T2(3)
     *build hours
5380 T1(2)=10*T2(4)+T2(5)
     *build minutes
5390 T1(3)=10*T2(6)+T2(7)
     *build seconds
5400 T1(4)=10*T2(8)+T2(9)
     *return
5410 RETURN
5500 REM SUBROUTINE TO TRANSMIT DATA TO "SMART" SYSTEM
     *open file #3 (rs232 output to "smart" system)
5510 OPEN "TV:" FOR OUTPUT AS FILE #3
```

```
     *transmit start char
5520 PRINT #3,"A"
     *set loop and transmit time
5530 FOR I=0 TO 4 \ PRINT #3,T1(I) \ NEXT I
     *transmit header data
     *output orbit count, phase, and charge counters
5540 PRINT #3,01 \ PRINT #3,Y1 \ PRINT #3,C2 \ PRINT #3,C4
     *set up batt loop
5550 FOR I=1 TO 6
     *output batt. no.
5560 PRINT #3,I
     *output cell volt.
5570 FOR J=1 TO 23 \ PRINT #3,D1(I,J) \ NEXT J
     *output cell press
5580 FOR J1=1 TO 23 \ PRINT #3,D8(I,J1) \ NEXT J1
     *output bat v & bat I
5590 PRINT #3,D2(I) \ PRINT #3,D3(I) \ PRINT #3,D9(I)
     *output temps
5600 FOR J=1 TO 6 \ PRINT #3,D5(I,J) \ NEXT J
     *output recond flags
5610 PRINT #3,V3(I)
     *get another
5620 NEXT I
     *output sas currents
5630 FOR I=1 TO 13 \ PRINT #3,D6(I) \ NEXT I
     *output bus data
5640 FOR J=1 TO 3 \ PRINT #3,A8(J) \ PRINT #3,A9(J) \ NEXT J
     *close file
5650 CLOSE #3
     *check for mag tape record
5660 IF M1=1 GOTO 5880
     *check skip orbit
5670 IF S2<0 GOTO 5880
     *check skip scan
5680 IF F1(9)=0 THEN S3=0 \ GOTO 5710
     *check for skip
5690 IF S3=0 GOTO 5710
     *reset skip flag
5700 S3=-1 \ GOTO 5870
     *increment record counter
5710 T9=T9+1
     *check for too many records
5720 IF T9<3700 GOTO 5740
     *too many print message
5730 PRINT "OUT OF TAPE" \ GOTO 5880
     *print start char
5740 PRINT #9,"A"
     *record time
5750 FOR I=0 TO 4 \ PRINT #9,T1(I) \ NEXT I
     *record header data
5760 PRINT #9,01 \ PRINT #9,P2 \ PRINT #9,C2 \ PRINT #9,C4
     *set up for battery loop
5770 FOR I=1 TO 6
     *record batt no.
```

```
5780 PRINT #9,I
     *set up loop and record cell volt
5790 FOR J=1 TO 23 \ PRINT #9,D1(I,J) \ NEXT J
     *set up loop and record cell press
5800 FOR J1=1 TO 23 \ PRINT #9,D8(I,J1) \ NEXT J1
     *record batt data
5810 PRINT #9,D2(I) \ PRINT #9,D3(I) \ PRINT #9,D9(I)
     *record batt temps
5820 FOR J=1 TO 6 \ PRINT #9,D5(I,J) \ NEXT J
     *record recond. flag
5830 PRINT #9,V3(I)
     *get another batt
5840 NEXT I
     *set up loop to record sas data
5850 FOR I=1 TO 13 \ PRINT #9,D6(I) \ NEXT I
     *set up loop for bus data
5860 FOR J=1 TO 3 \ PRINT #9,A8(J) \ PRINT #9,A9(J) \ NEXT J
     *add to skip counter
5870 S3=S3+1
     *return
5880 RETURN
6000 REM ORBDAT SUBROUTINE TO PRINT OUT ORBIT DATA
     *print first part of 1st line
6010 PRINT #1,"BAT  ORBIT      SUNSET TIME    BAT WHO
     BAT EFF  DCH  CHG    RR  ";
     *print secon part of 1st line
6020 PRINT #1,"   DOD    AHO   BPRC AH  CHI T-CAV T-CLO T--
     DHI T-DAV T-DLO T"
     *print first part of 1st data line
6030 PRINT #1,USING " #   #####  ####:###:##:##:##",I,01-1,
     T3(I,0),T3(I,1),T3(I,2),T3(I,3),T3(I,4);
     *check for zeros & set to limit valves
6040 IF A1(I)=0 THEN A1(I)=10000
6050 IF A2(I)=0 THEN A2(I)=1
6060 IF W1(I)=0 THEN W1(I)=1
6070 IF W2(I)=0 THEN W2(I)=1
     *calculate data
6080 G1=-1*W2(I)/60 \ G2=-1*W2(I)/W1(I) \ G3=A1(I)/(-1*A2(I))
6082 IF F3=0 THEN A4(I)=0
6084 G5=(-1*A2(I)/33)+A4(I)
6086 A4(I)=G5-(A1(I)/(R2*33)) \ G8=A4(I)
     *calculate data
6090 G6=-1*A2(I)/60 \ G7=A3(I)/60
6092 R9(I-1,J9)=G3 \ V=0
6094 FOR I=0 TO 9 \ V=V+R9(I-1,I2) \ NEXT I2 \ R=V/10
     *print data
6100 PRINT #1 USING "  ####.#     #.###",G1,G2; \ PRINT #1,
     TAB(49),G4;TAB(54);C2;" ";
     *print data
6110 PRINT #1,USING "##.###  ##.##  ##.##  ###.##",G3,G5,G6
     ,G7;
     *print data
6120 PRINT #1,USING "   ##.## ##.## ##.##  ##.## ##.## ##.##"
     ,E7(I),E9(I),E8(I),B7(I),B9(I),B8(I)
```

```
     *print first part of second header
6130 PRINT #1,"DHI CV--CNO--AV CV--DLO CV--CNO";
     *print 2nd part of 2nd header
6140 PRINT #1,TAB(40);"CHI CV--CNO--AV CV--CLO CV--CNO";
     \ B5=B6(I)/23 \ E5=E6(I)/23
6150 PRINT #1,TAB(80);"HI PRESS--CNO     DODD   SOC     SOD
     TTT  RRAV:
     *print the data
6061 PRINT #1, USING " #.###   ##  #.###   #.###   ##",B1(I),
     B2(I),B5,B3(I),B4(I);
6170 PRINT #1,TAB(40); \ PRINT #1,USING " #.###   ##  #.###
     #.### ##",E1(I),E2(I),E5,E3(I),E4(I);
6180 PRINT #1,TAB(80); \PRINT #1,USING "###.#     ##     ##.#
#   ##.##   ##.##  ###  #.###",A6(I),A7(I),A4(I),S6(I),S7(I),
T8,R
     *print space
6190 PRINT #1
     *skip if batt in recond
6200 IF V3(I)=1 GOTO 6230
     *clear sums
6205 FOR J=1 TO 6 \ S4(J)=S6 \ NEXT J
6210 A1(I)=0 \ A2(I)=0 \ W1(I)=0 \ W2(I)=0 \ A3(I)=0
6220 E1(I)=0 \ B3(I)=2 \ A6(I)=0
     *return
6230 RETURN
7000 REM ROUTINE TO CHECK HI-LO LIMITS
     *set limit value
7010 L9=L1
     *set up batt loop
7020 FOR I=1 TO 6
     *check for limit
7030 IF I>3 THEN L9=L2
     *set up cell loop
7040 FOR I1=1 TO 23
     *check limit, if out go shutdown
7050 IF D1(I,I1)>L9 GOTO 7140
     *check lo limit
7060 IF D1(I,I1)<L8 GOTO 7140
     *next cell
7070 NEXT I1
     *check batt curr
7080 IF D3(I)<-30 GOTO 7120
     *check batt curr hi
7090 IF D3(I)>25 GOTO 7120
     *get another batt
7100 NEXT I
     *return
7110 RETURN
     *go shutdown
7120 GOSUB 1790
     *print message
7130 PRINT #2,"BATTERY NO.";I;"CURRENT=";D3(I) \ CLOSE #2 \
     GOTO 7100
     *go shutdown
```

```
7140 GOSUB 1790
     *print message
7150 PRINT #2,"BATT=";I;"CELL=";I1;"V=";D1(I,I1) \ CLOSE #2 \
     GOTO 7070
7600 FOR I=1  TO 13
7610 U2=0
     *check limit
7620 IF D6(I)<5 THEN V5(I)=0 \ GOTO 7700
     *check if new out of limit
7630 IF V5(I)=1 GOTO 7700
7640 CALL CRATE(5) \ IF U2>2 GOTO 7680
7650 CALL C(8,0,16,(578+I))
     *check LAM
7651 CALL C(8,0,27,,Q,X)
     *if not ready try again
7652 IF Q=0 GOTO 7651
7660 CALL C(8,1,0,Z1%) \ D6(I)=Z1%*78.125 * 10^(-3)
7670 U2=U2+1 \ GOTO 7620
     *go alarm
7680 GOSUB 1800
     *print message
7690 PRINT #2,"SUPPLY NO.";I;"CURR=";D6(I) \ CLOSE #2 \
     V5(I) =1
     *get another
7700 NEXT I
7710 RETURN
     *special night routine
     *check flag
7800 IF P8=0 GOTO 7870
     *over 5 min. into phase
7810 IF C2>5 GOTO 7870
     *set up loop
7820 FOR I=1 TO 13
     *check limit
7830 IF D6(I)>5 GOTO 7860
     *out of limit go alarm
7840 GOSUB 1800
     *print message
7850 PRINT #2,"SUP #";I;"I=";D6(I) \ CLOSE #2
     *get another
7860 NEXT I
     *return
7870 RETURN
8200 REM SUBROUTINE TO GET HI-LO CELLS DURING DISCHARGE
     *set flag
8210 U=0
     *set up loop
8220 FOR J=1 TO 23
     *compare if lower skip to replace
8230 IF B3(I)<=D1(I,J) GOTO 8250
     *replace with new valve & set replace cell no. and set
     new hi
8240 B3(I)=D1(I,J) \ B6(I)=0 \ B4(I)=J \ U=1
     *if higher get next
```

```
8250 NEXT J
     *check flag if 0 skip
8260 IF U=0 GOTO 8320
     *set limit
8270 B1(I)=0
     *set loop
8280 FOR J1=1 TO 23
     *if less than high skip
8290 B6(I)=B6(I)+D1(I,J) \ IF B1(I)>D1(I,J1) GOTO 8310
     *replace high
8300 B1(I)=D1(I,J1) \ B2(I)=J1
     *get another
8310 NEXT J1
8320 RETURN
8350 REM SUBROUTINE TO GET HI-LO CELLS DURING CHARGE
8360 U=0
     *set up loop
8370 FOR J=1 TO 23
     *reset "try" counter
8380 U2=0
     *check for higher cell if yes skip to replace
8390 IF E1(I)>=D1(I,J) GOTO 8410
     *replace hi cell and batt v & get cell no. v
8400 E1(I)=D1(I,J) \ E6(I)=0 \ E2(I)=J \ U=1
     *get another
8410 NEXT J
8420 IF U=0 GOTO 8480
     *set lo cell
8430 E3(I)=2
     *set up loop
8440 FOR J1=1 TO 23
     *compare for lows
8450 E6(I)=E6(I)+D1(I,J1) \ IF E3(I)<D1(I,J1) GOTO 8470
     *replace valves
8460 E3(I)=D1(I,J1) \ E4(I)=J1
     *get another
8470 NEXT J1
8480 RETURN
8500 REM SUBROUTINE TO GET HI-LO TEMPS
     *set up limit
8510 H=-15 \ L=25 \ N=0
     *set up loop
8520 FOR J=1 TO 6
     *get sums
8530 N=N+D5(I,J)
     *check for new lo
8540 IF L>D5(I,J) THEN L=D5(I,J)
     *check for new hi
8550 IF H<D5(I,J) THEN H=D5(I,J)
     *get another
8560 NEXT J
     *compute average
8570 A=N/6
     *check for out of limit average
```

```
8580 IF A>25 GOTO 8610
     *check for lo out of limit
8590 IF A<-5 GOTO 8610
8600 RETURN
     *chamber shutdown and alarm
8610 CALL C(5,0,16,(2^13)) \ GOSUB 1790
     *print message
8620 PRINT #2,"BATTERY";I;"TEMP=";A \ CLOSE #2 \ GOTO 8600
9999 END
```

APPENDIX E

```
          NAME      ST6BINT
          SECTION   ST6B2
          LIST      DBG

;***********************************************
;***********************************************
;*****                                         *****
;*****    SPACE TELESCOPE                      *****
;*****    SIX BATTERY TEST                     *****
;*****    INTERRUPT HANDLER ROUTINE            *****
;*****                                         *****
;*****    BY:  JOHN R. BUSH, JR.               *****
;*****                                         *****
;*****    LATEST VERSION:  02/09/88            *****
;*****                                         *****
;***********************************************
;***********************************************
CR        EQU       ODH
LF        EQU       OAH
CU        EQU       15H
EOT       EQU       04H
SP        EQU       20H
TTYBASE   EQU       00C0H
;
          GLOBAL    TTYINT,TTYSETUP,WP2
```

```
;
;INTERRUPT ENTRY POINT
;
TTYINT   TB       21                ;IF SET, RECEIVED A CHAR
         JEQ      RCHAND            ;GO TO RECEIVER HANDLER
TXHAND   MOVB     @TXBUFF(R4),RO    ;GET NEXT CHAR TO XMIT
         SRL      RO,8
         CI       RO,EOT            ;IS IT THE END?
         JEQ      LASTCHAR          ;YES, JUMP
         CI       RO,3CH            ;< INDICATES MORE DATA
                                     ON "D" COMMAND
         JEQ      MORED
         INC      R4                ;NO POINT TO NEXT CHAR
         BL       @TXCHAR           ;TRANSMIT RO
         RTWP                       ;THROUGH THIS TIME
LASTCHAR CLR      RO                ;NULL
         CLR      R4
         BL       @TXCHAR           ; TO CLEAR INTERRUPT
         SBZ      19                ; DISABLE 9902 INTERRUPT
         RTWP
TXCHAR   SBO      16                ; TURN ON XMITTER
TXRDY    TB       22                ; SEE IF XMITTER READY
         JNE      TXRDY             ; NO, JUMP
         SLA      RO,8              ; SHIFT FOR LDCR
         LDCR     RO,8              ; SEND CHAR
         SBZ      16                ; TURN OFF XMITTER
         B        *R11              ;RETURN
;
MORED    B        MOREDISP          ;GET IN RANGE
;
;
;
;CONVERT RO TO ASCII
;
TASCII   AI       RO,30H            ;FOR ALL ASCII
         CI       RO,3AH            ; A - F?
         JLT      NOCONV            ; NO, OK
         AI       RO,7              ; FOR LETTERS
NOCONV   B        *R11
;
;
```

```
;
; IF JUST RECEIVED A CHAR YOU ARE HERE
;
RCHAND   STCR    RO,8              ; GET IT
         SRL     RO,8              ; SHIFT TO LSB
         SBO     18                ; RESET INTERRUPT
         ANDI    RO,007FH          ; STRIP PARITY
         CI      RO,CU             ; IS IT A CONTROL U?
         JEQ     CUTASK            ; GO DO IT
         CI      RO,CR             ; IS IT A CR?
         JEQ     CRTASK            ; GO DO IT
         BL      @TXCHAR           ; ELSE, ECHO CHARACTER
                                   ; TXCHAR ALSO SHIFTS SO
                                     DONT NEED
                                   ; NEXT STEP HERE....
         ;SLA    RO,8              ; SHIFT FOR BYTE STORE
                                     (HERE AND ELSEWHERE)
         MOVB    RO,@RCBUFF(R3)    ; STORE CHAR
         INC     R3                ; INC RCBUFF POINTER
         RTWP
;
; CONTROL U TASK
;
CUTASK   CLR     R3
         CLR     R4
         MOVB    @CUMSG(R4),@TXBUFF(R4)
         INC     R4
         MOVB    @CUMSG(R4),@TXBUFF(R4)
         INC     R4
         BL      @PROMPT
         RTWP
;
; CR TASK ... HERE GOES
;
CRTASK   SLA     RO,8
         MOVB    RO,@RCBUFF(R3)    ; STORE CR
FDCS     CLR     R3
         CLR     R4
         MOVB    @RCBUFF (R3),RO
         INC     R3
         SRL     RO,8
         CI      RO,"X"            ; X?
         JEQ     XTASK
         CI      RO,"S"            ; S?
         JEQ     STASK
         CI      RO,"D"            ; D?
         JEQ     DTASK
PQTASK   BL      @QTASK            ; NONE ON THE ABOVE,
                                     ABORT
         RTWP
```

```
;
; X COMMAND ROUTINE
;
XTASK    LI      R1,XTABE      ; X TABLE END ADDRESS
         LI      R2,XTAB       ; X TABLE START ADDRESS
         BL      @LDPAR        ; GET PARAMETER ADDRESS
         MOV     RO,RO         ; CHECK FOR ERROR
         JEQ     PQTASK        ; NOT SUCCESSFUL, ABORT
         MOV     *RO,R1        ; GET DATA
         MOV     R1,RO         ; COPY TO RO
         BL      @NEWLINE      ; PUT CR,LF INTO TXBUFF
         BL      @CONTBUFF     ; CONVERT TO ASCII, THEN
                               TO TXBUFF
         BL      @PROMPT       ; ADD A PROMPT
         RTWP                  ; AND RETURN
;
; S TASK COMMAND ROUTINE
;
STASK    LI      R1,STABE      ; S TABLE END ADDRESS
         LI      R2,STAB       ; S TABLE START ADDRESS
         LI      R6,5          ; ABORT POSITION FOR CR
                               AT WRONG PLACE
         BL      @LDPAR        ; GET PARAMETER ADDRESS
         MOV     RO,RO         ; SEE IF CLEAR
         JEQ     PQTASK        ; UNSUCCESSFUL, ABORT
         BL      @CKDEL        ; GOOD DELIMITER?
         MOV     R1,R1         ; SEE IF CLEAR
         JEQ     PQTASK        ; NO, ABORT
         BL      @GDATAAD      ; TO GET DATA FROM
                               RCBUFF
         MOV     R1,R1         ; SEE IF CLEAR
         JEQ     PQTASK        ; YES, NOT SUCCESSFUL,
                               ABORT
         MOV     R2,*RO        ; STORE NEW DATA,
         CLR     R4            ; AND
         BL      @PROMPT       ; GET
         RTWP                  ; OUT
```

```
;
; D TASK COMMAND ROUTINE
;
DTASK    LI      R6,2            ; ABORT POSITION FOR CR
         CLR     R7
         CLR     R8
         BL      @GDATAAD        ; GET START ADDR
         MOV     R1,R1           ; SEE IF CLEAR
         JEQ     PQTASK          ; YES, ABORT
         LI      R7,1            ; MAKE START ADDRESS TO
         SZC     R7,R2           ; BE THE NEXT LOWEST
                                   ADDRESS
         MOV     R2,R7           ; IF IT WAS ODD. SAVE IN
                                   R7.
         CI      R9,ODOOH        ; WAS LAST CHAR A CR?
         JNE     COND            ; NO
SDADD    BL      @NEWLINE        ; START A NEWLINE
         MOV     R7,RO
         BL      @CONTBUFF       ; PUT ASCII IN TXBUFF
         BL      @ADSPCE         ; ADD A SPACE
         MOV     *R7,RO          ; GET DATA
         BL      @CONTBUFF       ; TO TXBUFF
         BL      @PROMPT         ; ADD A PROMPT
         RTWP                    ; GET OUT
;
; CHECK FOR SECOND ADDRESS
;
COND     BL      @GDATAAD        ; GET END ADDRESS
         MOV     R1,R1           ; SEE IF CLEAR
         JEQ     PQTASK          ; ABORT
         LI      R8,1            ; MAKE END ADDRESS TO BE
                                   NEXT
         SZC     R8,R2           ; LOWEST ADDRESS IF IT
                                   WAS ODD.
         MOV     R2,R8           ; SAVE END ADDRESS
         MOV     R7,R2           ; START ADDR
         C       R2,R8           ; MAKE SURE END IS >
                                   THAN START
         JHE     SDADD           ; IT WAS NOT, SO JUST DO
                                   ONE
```

```
;
; IF HERE, START LOADING TXBUFF ONE LINE AT A TIME, ENDING
; THE LINE ON XXXEH WITH THE LAST CHARACTER BEING A "<" (3CH)
; IF DTASK IS NOT FINISHED.  START NEW LINES ON XXXOH.
; WHEN LAST DATA IS LOADED, DO "BL @PROMPT".
;
MOREDISP BL       @NEWLINE          ; START A NEW LINE
         MOV      R7,RO
         BL       @CONTBUFF         ; LOAD ADDR IN TXBUFF
MOREDL   BL       @ADSPCE           ; ADD A SPACE
         MOV      *R7,RO            ; LOAD DATA
         BL       @CONTBUFF         ; DATA TO TXBUFF
         C        R7,R8             ; CHECK FOR END
         JHE      DEND
         MOV      R7,RO
         ANDI     RO,OFH            ; STRIP ALL BUT LSN
         CI       RO,OEH            ; IS IT AN E?
         JEQ      ADLA              ; YES, ADD LEFT ARROW
         INCT     R7                ; ELSE, INC TO NEXT
                                      ADDRESS
         JMP      MOREDL            ; AND DO AGAIN
ADLA     LI       RO,3COOH
         MOVB     RO,@TXBUFF(R4)
         INCT     R7
         CLR      R4
         SBO      19                ;ENABLE 9902 INTERRUPT
         RTWP
DEND     BL       @PROMPT
         RTWP
```

```
;
; SUBROUTINE TO LOAD A PARAMETER FROM RCBUFF
; SEARCH TABLE STARTING AT R2, ENDING AT
; R1 AND RETURN ADDRESS OF PARAMETER IN RO.
; IF INPUT WAS JUST XC, WILL SUBSTITUTE D8040,805E
; IN ORDER TO DISPLAY ALL ANALOG INPUT CHANNELS.
; IF UNSUCCESSFUL, RETURNS OS IN RO.
;
LDPAR   CLR     RO
        CLR     R5
        MOVB    @RCBUFF(R3),RO      ; FIRST CHAR
        INC     R3
        MOVB    @RCBUFF(R3),R5      ; SECOND CHAR
        INC     R3
        SRL     R5,8
        A       R5,RO               ; RO HAS PARAMETER
        CI      RO,430DH            ; WAS IT JEST XC?
        JEQ     DCS
FPARL   C       RO,*R2+             ; MATCH?
        JEQ     FPAR                ; YES
        INCT    R2                  ; NO, SKIP OVER ADDRESS
        C       R2,R1               ; END OF TABLE?
        JHE     NPAR                ; YES, NO MATCH
        JMP     FPARL               ; NO MATCH YET, TRY
                                      AGAIN
FPAR    MOV     *R2,RO              ; LOAD ADDRESS OF
                                      PARAMETER
        B       *R11
NPAR    CLR     RO                  ; NO MATCH, RETURN ZERO
        B       *R11
DCS     CLR     R3
DCSL    MOVB    @DDCS(R3),@RCBUFF(R3)
        MOVB    @DDCS(R3),RO
        INC     R3
        SRL     RO,8
        CI      RO,CR
        JNE     DCSL
        B       FDCS
;
; SUBROUTINE TO CHECK NEXT RCBUFF CHAR FOR A VALID
;DELIMITER (BLANK OR COMMA), RETURNS ZERO IN R1 IF
;NOT SUCCESSFUL.
;
CKDEL   CLR     R1
        MOVB    @RCBUFF(R3),R1
        INC     R3
        CI      R1,2000H            ; SPACE?
        JEQ     GDDEL               ; YES
        CI      R1,2COOH            ; COMMA?
        JEQ     GDDEL               ; YES
        CLR     R1                  ; NOT SUCCESSFUL
GDDEL   B       *R11                ; BRANCH OUT
```

```
;
; SUBROUTINE TO GET DATA FROM RCBUFF, R3 IS OFFSET,
; SEARCH RCTABLE UNTIL FIND GOOD DELIMITER OR CR,
; ROTATING GOOD DATA INTO R2;  IF FIND BAD ASCII CHAR,
; RETURN ZEROS IN R1.  DO NOT USE RO !!
;
GDATAAD  MOV     R11,R10          ; NESTED SUBROUTINE
                                    LINKAGE
         CLR     R2
GDTL     CLR     R1
         MOVB    @RCBUFF(R3),R1
         INC     R3
         MOV     R1,R9            ; SAVE FOR DTASK
         CI      R1,2000H         ; SPACE?
         JEQ     FDELCR           ; YES
         CI      R1,2COOH         ; COMMA?
         JEQ     FDELCR           ; YES
         CI      R1,ODOOH         ; CR?
         JEQ     FDELCR           ; YES
         BL      @FRASCII         ; NO, TEN WHAT IS IT?
         MOV     R5,R5            ; GOOD DATA?
         JEQ     BDDATA           ; NO, GET OUT
         SLA     R2,4             ; YES
         A       R1,R2            ; ADD INTO R2
         JMP     GDTL             ; AND DO UNTIL DELIMITER
                                    OR CR
FDELCR   C       R3,R6            ; IF THIS HAPPENS AT
                                    CHAR + R6
         JEQ     BDDATA           ; THEN NOTHING WAS
                                    THERE, ABORT.
         SETO    R1               ; SIGNAL GOOD DATA
         B       *R10
BDDATA   CLR     R1               ; SIGNAL BAD DATA
         B       *R10
FRASCII  SRL     R1,8             ; CONVERT FROM ASCII
         AI      RI,-30H          ; SUBTRACT 30H
         JLT     NOTAC            ; IF LESS THAN ZERO,
                                    ABORT
         CI      R1,10            ; SEE IF FINISHED
         JLT     GDAC             ; YES
         AI      R1,-7            ; FOR LETTERS
         JLT     NOTAC            ; ABORT
         CI      R1,OFH           ; BIGGER THAN F?
         JGT     NOTAC            ; YES, YOU GUESSED IT
                                    -- ABORT
GDAC     SETO    R5               ; SIGNAL GOOD DATA
         B       *R11
NOTAC    CLR     R5
         B       *R11
```

```
;
; SUBROUTINE TO CONVERT RO TO ASCII AND STORE
; IN TXBUFF.  DESTROYS RO
;
CONTBUFF MOV       R11,R10          ; FOR NESTED SUBROUTINE
                                      LINKAGE
         MOV       RO,R1            ; COPY DATA
         SRL       RO,12            ; SHIFT TO FIRST CHAR
         BL        @BUFFSUB         ; CONVERT AND STORE
         SRL       RO,8
         BL        @BUFFSUB
         SRL       RO,4
         BL        @BUFFSUB
         BL        @BUFFSUB
         B         *R10
BUFFSUB  MOV       R11,R9           ; FOR NESTED LINKAGE
         ANDI      RO,OFH           ; STRIP ALL BUT LSN
         BL        @TASCII          ; TO ASCII
         SLA       RO,8
         MOVB      RO,@TXBUFF(R4)
         INC       R4
         MOV       R1,RO
         B         *R9
;
; SUBROUTINE TO PUT CR,LF INTO TXBUFF
;
NEWLINE  LI        R1,ODOOH
         CLR       R4
         MOVB      R1,@TXBUFF(R4)
         INC       R4
         LI        R1,OAOOH
         MOVB      R1,@TXBUFF(R4)
         INC       R4
         B         *R11
;
; SUBROUTINE TO ADD SPACE TO TXBUFF
;
ADSPCE   LI        R1,2000H
         MOVB      R1,@TXBUFF(R4)
         INC       R4
         B         *R11
```

```
;
; Q TASK
;
QTASK     CLR       R4
QTLP      MOVB      @QMSG(R4),@TXBUFF(R4)
          MOVB      @TXBUFF(R4),RO
          SRL       RO,8
          INC       R4
          CI        RO,"?"
          JNE       QTLP
PROMPT    CLR       R3
PRLP      MOVB      @PMSG(R3),@TXBUFF(R4)          ;MOVE CHAR
          MOVB      @TXBUFF(R4),RO     ; CHECK FOR EOT
          SRL       RO,8
          INC       R3
          INC       R4
          CI        RO,EOT
          JNE       PRLP
          CLR       R3
          CLR       R4
          SBO       19                 ; ENABLE 9902 INTERRUPT
          B         *R11
```

```
;
; TABLE OF X PARAMETERS
;
XTAB      ASCII     "CO"
          WORD      8040H
          ASCII     "C1"
          WORD      8042H
          ASCII     "C2"
          WORD      8044H
          ASCII     "C3"
          WORD      8046H
          ASCII     "C4"
          WORD      8048H
          ASCII     "C5"
          WORD      804AH
          ASCII     "C6"
          WORD      804CH
          ASCII     "C7"
          WORD      804EH
          ASCII     "C8"
          WORD      8050H
          ASCII     "C9"
          WORD      8052H
          ASCII     "CA"
          WORD      8054H
          ASCII     "CB"
          WORD      8056H
          ASCII     "CC"
          WORD      8058H
          ASCII     "CD"
          WORD      805AH
          ASCII     "CE"
          WORD      805CH
          ASCII     "CF"
          WORD      805EH
          ASCII     "DE"
          WORD      8092H
          ASCII     "TD"
          WORD      8094H
          ASCII     "R1"
          WORD      8060H
          ASCII     "R2"
          WORD      8062H
          ASCII     "R3"
          WORD      8064H
          ASCII     "A1"
          WORD      8066H
          ASCII     "A2"
          WORD      8068H
          ASCII     "A3"
          WORD      806AH
          ASCII     "HI"
          WORD      806CH
```

```
ASCII     "LI"
WORD      806EH
ASCII     "H2"
WORD      8070H
ASCII     "L2"
WORD      8072H
ASCII     "H3"
WORD      8074H
ASCII     "L3"
WORD      8076H
ASCII     "H4"
WORD      8078H
ASCII     "L4"
WORD      807AH
ASCII     "H5"
WORD      807CH
ASCII     "L5"
WORD      807EH
ASCII     "H6"
WORD      8080H
ASCII     "L6"
WORD      8082H
ASCII     "UH"
WORD      8084H
ASCII     "UL"
WORD      8086H
ASCII     "LH"
WORD      8088H
ASCII     "LL"
WORD      808AH
ASCII     "PI"
WORD      808CH
ASCII     "P2"
WORD      808EH
ASCII     "P3"
WORD      8090H
ASCII     "M1"
WORD      8096H
ASCII     "M2"
WORD      8098H
ASCII     "M3"
WORD      809AH
ASCII     "I1"
WORD      809CH
ASCII     "I2"
WORD      809EH
ASCII     "I3"
WORD      80A0H
ASCII     "PS"
WORD      80A4H
ASCII     "ES"
WORD      80A6H
ASCII     "TR"
```

```
        WORD    8048H
        ASCII   "NS"
        WORD    80AAH
        ASCII   "KI"
        WORD    80B0H
        ASCII   "PO"
        WORD    80A2H
        ASCII   "CT"
        WORD    80B2H
        ASCII   "SO"
        WORD    80BAH
        ASCII   "TS"
        WORD    80BCH
        ASCII   "PL"
        WORD    80BEH
        ASCII   "ET"
        WORD    80C4H
        ASCII   "NR"
        WORD    80CAH
        ASCII   "PR"
        WORD    80CCH
        ASCII   "EP"
        WORD    80D4H
        ASCII   "D1"
        WORD    80D6H
        ASCII   "D2"
        WORD    80D8H
        ASCII   "D3"
        WORD    80DAH
        ASCII   "D4"
        WORD    80DCH
        ASCII   "D5"
        WORD    80DEH
        ASCII   "D6"
        WORD    80E0H
XTABE   EQU     $
```

TABLE OF S PARAMETERS

```
STAB    ASCII   "DE"
        WORD    8092H
        ASCII   "TD"
        WORD    8094H
        ASCII   "A1"
        WORD    8066H
        ASCII   "A2"
        WORD    8068H
        ASCII   "A3"
        WORD    806AH
        ASCII   "HI"
        WORD    806CH
        ASCII   "L1"
        WORD    806EH
        ASCII   "H2"
        WORD    8070H
        ASCII   "L2"
        WORD    8072H
        ASCII   "H3"
        WORD    8074H
        ASCII   "L3"
        WORD    8076H
        ASCII   "H4"
        WORD    8078H
        ASCII   "L4"
        WORD    807AH
        ASCII   "H5"
        WORD    807CH
        ASCII   "L5"
        WORD    807EH
        ASCII   "H6"
        WORD    8080H
        ASCII   "L6"
        WORD    8082H
        ASCII   "UH"
        WORD    8084H
        ASCII   "UL"
        WORD    8086H
        ASCII   "LH"
        WORD    8088H
        ASCII   "LL"
        WORD    808AH
        ASCII   "P1"
        WORD    808CH
        ASCII   "P2"
        WORD    808EH
        ASCII   "P3"
        WORD    8090H
        ASCII   "M1"
        WORD    8096H
```

```
            ASCII       "M2"
            WORD        8098H
            ASCII       "M3"
            WORD        809AH
            ASCII       "I1"
            WORD        809CH
            ASCII       "I2"
            WORD        809EH
            ASCII       "I3"
            WORD        80A0H
            ASCII       "PS"
            WORD        80A4H
            ASCII       "ES"
            WORD        80A6H
            ASCII       "TR"
            WORD        80A8H
            ASCII       "NS"
            WORD        80AAH
            ASCII       "KI"
            WORD        80B0H
            ASCII       "TS"
            WORD        80BCH
            ASCII       "PL"
            WORD        80BEH
            ASCII       "ET"
            WORD        80C4H
            ASCII       "NR"
            WORD        80CAH
            ASCII       "PR"
            WORD        80CCH
            ASCII       "EP"
            WORD        80D4H
            ASCII       "D1"
            WORD        80D6H
            ASCII       "D2"
            WORD        80D8H
            ASCII       "D3"
            WORD        80DAH
            ASCII       "D4"
            WORD        80DCH
            ASCII       "D5"
            WORD        80DEH
            ASCII       "D6"
            WORD        80E0H
STABE       EQU         $
;
QMSG        BYTE        CR,LF,"?",CR
PMSG        BYTE        CR,LF,">",EOT
CUMSG       ASCII       "^U"
DDCS        ASCII       "D8040 805E"

            BYTE        CR,0
```

```
TTYSETUP WORD     WP2
         WORD     $+2
         CLR      R3
         CLR      R4
         LI       R12,TTYBASE
         BL       @PROMPT
         RTWP
ZZZ      EQU      $
;
;
         SECTION  INTHRWM,ABSOLUTE
         ORG      8100H
RCBUFF   BLOCK    16
TXBUFF   BLOCK    50
         END
```

```
                NAME      ST6BTASK
                TITLE     "ST6B TASKS"
                SECTION   ST6B1,ABSOLUTE
                LIST      ME,DBG
                GLOBAL    TTYINT,TTYSETUP,WP2
                EQU       8000H
        RWM
        ;
        ;***********************************************
        ;***********************************************
        ;*****                                     *****
        ;*****                                     *****
        ;*****     SPACE TELESCOPE                 *****
        ;*****     SIX BATTERY TEST                *****
        ;*****     CONTROLLER                      *****
        ;*****                                     *****
        ;*****     BY:  JOHN R. BUSH, JR.          *****
        ;*****                                     *****
        ;*****     LATEST VERSION:  02/09/88       *****
        ;*****                                     *****
        ;*****                                     *****
        ;***********************************************
        ;***********************************************
;
;
; ASSIGN INTERRUPT VECTORS
;
        ORG       0
        WORD      WP1                 ; FIRST WORKSPACE
        WORD      START               ; ST6B SOFTWARE START
        WORD      WP2                 ; TTY INTERRUPT
                                        WORKSPACE
        WORD      TTYINT              ; TTY INTERRUPT
                                        SOFTWARE START
;
; BEGIN ST6B SOFTWARE
;
        ORG       80H
;
; SET UP A TO D
;
ATOD    EQU       0
ADEOC   EQU       13                  ; END OF CONVERT
ADSC    EQU       24                  ; START CONVERT
SHMD    EQU       23                  ; MODE OF S/H
MPXEN   EQU       18                  ; MULTIPLEXER ENABLE
ADLSB   EQU       1                   ; A/D LSB
MPXAL   EQU       19                  ; MULTIPLEXER ADDRESS
;
;       START ST6B SOFTWARE
;
```

```
START    LIMI     0                    ; DISABLE INTERRUPTS
         LI       R12,ATOD             ; A/D BASE ADDRESS
         SBZ      MPXEN                ; ENABLE LOW
         SBO      ADSC                 ; START CONVERT HIGH
         SBZ      SHMD                 ; MODE LOW
;
; SET UP D TO A
;
DA1      EQU      0100H                ; BASE OF D/A 1
DA2      EQU      0140H                ; BASE OF D/A 2
DA3      EQU      0180H                ; BASE OF D/1 3
BVH      EQU      28                   ; CRU BIT BAT V HIGH
DALSB    EQU      16                   ; PORT OF LSB
PULSE    EQU      28                   ; PULSE BIT CRU
                                        POSITION
TFLAG    EQU      29                   ; TRIP FLAG (TR RELAYS
                                        HAVE TRIPPED)
V15      EQU      5                    ; 15 VOLT SENSE
;
         LI       R12,DA1              ; D/A BASE ADDRESS
         SBO      1                    ; ENABLE ONE INTERRUPT
                                        FOR TTY
         SBZ      PULSE                ; ZERO PULSE
         SBZ      TFLAG                ; ZERO TRIP FLAG
         SETO     RO                   ; ALL ONES
         LI       R12,DA1+(DALSB*2)    ; BASE FOR DIGITAL DATA
         LDCR     RO,12                ; OUTPUT ALL ONES TO
                                        D/A
         LI       R1,0700H             ; OTHER 2 D/AS
         LI       R12,DA2+(DALSB*2)
         LDCR     RO,12
         LI       R12,DA2+(BVH*2)      ; BUS OVERVOLTAGE
         LDCR     R1,4                 ; AND HEATER TAPE
         LI       R12,DA3+(DALSB*2)
         LDCR     RO,12
         LI       R12,DA3+(BVH*2)
         LDCR     R1,4
;
; SETUP VDT CONTROL (9902)
;
RTRATE   EQU      209                  ; RECEIVE AND TRANSMIT
                                        RATE
                                        ; (WITH 2 MHZ CLOCK)
CNTWD    EQU      0E200H               ; CONTROL WORD (MOST
                                        SIG BYTE)
TTYBASE  EQU      00C0H                ; ADDRESS OF 9902
;
;
         LI       R12,TTYBASE          ; 9902 BASE ADDRESS
         LI       RO,CNTWD
         LI       R1,RTRATE
         SBO      31                   ; RESETS 9902
         LDCR     RO,8                 ; OUTPUT CNT WD,
                                        RESETS LDCTRL
```

```
          SBZ       13                    ; NOTHING INTO INTERVAL
                                            REGISTER
          LDCR      R1,12                 ; DATA RATE, ALSO SETS
                                            ALL INTERNAL
                                          ; FLAGS...READY TO GO
          SBO       18                    ; ENABLES 9902 RCV INT
;
; SET UP CCC WATCHER AND POWER SUPPLY CONTROL
;
CCCIN     EQU       40H                   ; ADDRESS OF CCC INPUTS
POUT      EQU       80H                   ; ADDRESS OF POWER
                                            SUPPLY OUTPUTS
CCCLSB    EQU       1
POUTLSB   EQU       16
          LI        R12,POUT+(POUTLSB*2)  ;FOR OUTPUT, DONT
                                           NEED INPUT
          CLR       R1                    ; ALL 0S
          LDCR      R1,13
;
; INITIALIZE ALL PARAMETERS
;
DIN       EQU       1          ; INITIAL DELTA
ITD       EQU       88H        ; INITIAL TIME DELAY
H1I       EQU       0CA9H      ; INITIAL DIODE TRIP POINT HIGH
L1I       EQU       0C05H      ; INITIAL DIODE TRIP POINT LOW
H2I       EQU       0F99H      ; INITIAL DIODE TRIP POINT HIGH
L2I       EQU       0ECCH      ; INITIAL DIODE TRIP POINT LOW
H3I       EQU       0F99H      ; INITIAL DIODE TRIP POINT HIGH
L3I       EQU       0ECCH      ; INITIAL DIODE TRIP POINT LOW
H4I       EQU       0F99H      ; INITIAL DIODE TRIP POINT HIGH
L4I       EQU       0ECCH      ; INITIAL DIODE TRIP POINT LOW
H5I       EQU       0F99H      ; INITIAL DIODE TRIP POINT HIGH
L5I       EQU       0ECCH      ; INITIAL DIODE TRIP POINT LOW
H6I       EQU       0F99H      ; INITIAL DIODE TRIP POINT HIGH
L6I       EQU       0ECCH      ; INITIAL DIODE TRIP POINT LOW
UHI       EQU       2ABH       ; INITIAL PLATE TEMPS TOP HIGH
ULI       EQU       290H       ; UPPER LOW
LHI       EQU       2ABH       ; LOWER HIGH
LLI       EQU       290H       ; LOWER LOW
APRV1I    EQU       410        ; INITIAL GUESS FOR PRVS (10 A)
APRV2I    EQU       410
APRV3I    EQU       410
P1I       EQU       733        ; 731W INITIAL
P2I       EQU       717
P3I       EQU       733
M1I       EQU       0E66H
M2I       EQU       0E66H
M3I       EQU       0E66H
I1I       EQU       1229       ; 30 AMPS (LOAD BANK I, 15/BATT)
I2I       EQU       1229
I3I       EQU       1229
PSI       EQU       0          ; POWER SUPPLY CONTROL
TRI       EQU       1          ; NO. OF TRIPS BEFORE P.S.
                                   SHUTOFF
```

```
NSI       EQU       8          ; TOTAL NO. OF SPAS OFF AT
                                 TR=CCC TRIPS
TSI       EQU       0FFFH      ; MAX AVG TEMP ON PLATES
PLI       EQU       50         ; CYCLES BETWEEN PULSES
TRFI      EQU       0          ; TRIP BACK FLAG
DNFI      EQU       0          ; DAY NIGHT FLAG
ETI       EQU       14
ETCCI     EQU       100H
EPI       EQU       325
D1I       EQU       50         ; DELTA POWER FOR ADD FIRST TIME
                                 OUT
D2I       EQU       50         ; DELTA POWER FOR SUB SECOND
                                 TIME OUT
D3I       EQU       83         ; DELTA....
D4I       EQU       83
D5I       EQU       67
D6I       EQU       67
;
;
          MACRO     WINIT      ; MACRO DEFINITION
          LI        RO,'1'
          MOV       RO,@'2'
          ENDM
;
;
          WINIT     DIN,DE
          WINIT     ITD,TDLY
          WINIT     H1I,H1
          WINIT     L1I,L1
          WINIT     H2I,H2
          WINIT     L2I,L2
          WINIT     H3I,H3
          WINIT     L3I,L3
          WINIT     H4I,H4
          WINIT     L4I,L4
          WINIT     H5I,H5
          WINIT     L5I,L5
          WINIT     H6I,H6
          WINIT     L6I,L6
          WINIT     UHI,UH
          WINIT     ULI,UL
          WINIT     LHI,LH
          WINIT     LLI,LL
          WINIT     APRV1I,APRV1
          WINIT     APRV2I,APRV2
          WINIT     APRV3I,APRV3
          WINIT     P1I,P1
          WINIT     P2I,P2
          WINIT     P3I,P3
          WINIT     M1I,M1
          WINIT     M2I,M2
          WINIT     M3I,M3
          WINIT     I1I,I1
          MOV       RO,@I2
```

```
          MOV       R0,@I3
          WINIT     PSI,PS
          MOV       R0,@ES
          MOV       R0,@TRF
          MOV       R0,@DNF
          MOV       R0,@KI
          MOV       R0,@LBCNT
          MOV       R0,@PR
          MOV       R0,@EPC
          MOV       R0,@EPCC
          WINIT     TRI,TR
          MOV       R0,@NEWRF
          WINIT     EPI,EP
          WINIT     D1I,D1
          WINIT     D2I,D2
          WINIT     D3I,D3
          WINIT     D4I,D4
          WINIT     D5I,D5
          WINIT     D6I,D6
          WINIT     NSI,NS
          WINIT     TSI,TS
          WINIT     PLI,PL
          MOV       R0,@PLC
          WINIT     ETI,ET
          WINIT     ETCCI,ETCC
;
          BLWP      @TTYSETUP           ; GO SETUP TTY
                                          WORKSPACE AND
                                        ; PUT OUT PROMPT
          LIMI      1                   ; ENABLE INTERRUPTS
;
;
;         SAMPLE ROUTINE
;
SAMPLE    LI        R2,CHL0             ; CHANNEL 0 STORAGE
                                          ADDRESS
          LI        R1,0016             ; COUNTER
          CLR       R0                  ; ADDR OF MPX ANALOG
                                          INPUT
SAMPLOOP  LI        R12,MPXAL*2         ; BASE OF MPX ADDR
          SLA       R0,8                ; GET TO MSB FOR LDCR
          LDCR      R0,4                ; SEND OUT ADDR
          SRL       R0,8                ; GET BACK FOR INC
          LI        R12,ATOD            ; BASE
          LIMI      0                   ; MUST DISABLE
                                          INTERRUPTS
          SBO       MPXEN               ; ENABLE MPX
          SBO       SHMD                ; SAMPLE
          SBO       SHMD                ; WAIT
          SBZ       SHMD                ; HOLD
          SBZ       MPXEN               ; DISABLE MPX
          SBZ       ADSC                ; START CONVERSION
          SBO       ADSC                ; PROCESS
NOTEOC    TB        ADEOC               ; SEE IF THROUGH
```

```
          JEQ       NOTEOC              ;NO, WAIT
          LI        R12,ADLSB*2         ; BASE OF ADLSB
          STCR      R4,12; GET DIGITAL DATA
          LIMI      1                   ; ENABLE INTERRUPTS
          INV       R4                  ; A/D INVERTING TYPE
          ANDI      R4,OFFFH            ; STRIP LEADING ONES
                                          FROM INV
          MOV       R4,*R2+             ; STORE INVERTED DATA
          INC       R0                  ; INC ADDR OF MPX
          DEC       R1                  ; DEC COUNTER
          JNE       SAMPLOOP            ; DO 16 TIMES
; CORRECT CHLS 1-9 WITH CHLO (OFFSET).
;
          LI        R1,0
          LI        R2,CHL1
CLOOP     S         @CHLO,*R2
          JGT       NNEG
          CLR       *R2
NNEG      INCT      R2
          DEC       R1
          JNE       CLOOP
;
; FIRST OF ALL, SEE IF IN TRICKLE CHARGE/CAP TEST (CONSTANT
; CURRENT DISCHARGE MODE) (KI = 1). IF SO AND IN NIGHT
; THEN PRVS = I1,I2,I3. IF IN DAY THEN DO THIS...
; IF (VB1+VB2)/2>M1 THEN PRV1=2048 (APRVS INC SLOWLY IN
  OUTPRV)
; ELSE PRV1=0. SAME FOR 3-4, AND 5-6.
;
          MOV       @KI,@KI
          JEQ       CLOAD
          LI        R12,DA1
          TB        4                   ; DAY/NIGHT FLAG HERE
          JEQ       TRC
          MOV       @I1,@PRV1
          MOV       @I2,@PRV2
          MOV       @I3,@PRV3
          B         OUTPRV
TRC       LI        R2,2048
          MOV       @VB1,R1
          A         @VB2,R1
          SRL       R1,1                ; DIVIDE BY 2
          C         R1,@M1
          JLT       ZPRVI
          MOV       R2,@PRV1
          JMP       CKV3V4
ZPRV1     CLR       @PRV1
CKV3V4    MOV       @VB3,R1
          A         @VB4,R1
          SRL       R1,1
          C         R1,@M2
          JLT       ZPRV2
          MOV       R2,@PRV2
          JMP       CKV5V6
```

```
ZPRV2    CLR     @PRV2
CKV5V6   MOV     @VB5,R1
         A       @VB6,R1
         SRL     R1,1
         C       R1,@M3
         JLT     ZPRV3
         MOV     R2,@PRV3
         JMP     OUTPRV
ZPRV3    CLR     @PRV3
         JMP     OUTPRV
;
; CALCULATE V TO OUTPUT TO LOAD BANKS AT .1V/AMP
; P1,P2,P3 = EACH LB POWER DESIRED; VLB1,VLB2,VLB3=
; EACH LB VOLTAGE RANGED 0-40V (CHL1,2,3, 0-10V IN)
; EQUATION IS:
; P1(,2,3)*4095/VLB1(,2,3)*4095/4000 WHERE VLB1(,2,3)>
; 63 (MIN VOLTAGE FOR 32 BITS AT 1000W) AND FINAL
; ANSWER < 2048 (MAX INPUT TO LOAD BANKS)
;
CLOAD    LI      R3,4095
         LI      R4,4000
         MOV     @P1,R1          ; POWER DESIRED
         CI      R1,0
         JEQ     CON2
         MOV     @VLB1,R5        ; SAMPLED VOLTAGE
         BL      @CALC           ; TO CALCULATE
CON2     MOV     R1,@PRV1        ; STORE IT
         MOV     @P2,R1          ; ETC 3 TIMES
         CI      R1,0
         JEQ     CON3
         MOV     @VLB2,R5
         BL      @CALC
CON3     MOV     R1,@PRV2
         MOV     @P3,R1
         CI      R1,0
         JEQ     CON4
         MOV     @VLB3,R5
         BL      @CALC
CON4     MOV     R1,@PRV3
         JMP     OUTPRV
CALC     MOV     @TRF,R7
         JNE     CALCC
         MOV     @PR,R7
         JEQ     CALCC
         MOV     @EPCC,R7
         JEQ     TEPC
         DEC     @EPCC
         JMP     LBRT1
TEPC     MOV     @EPC,R7
         JEQ     LBRT
         DEC     @EPC
         JMP     LBRT2
LBRT     INC     @LBCNT
         MOV     @EP,@EPC
```

```
LBRT2    LI      R7,ETCCI
         MOV     R7,@EPCC
LBRT1    MOV     @LBCNT,R7
         CI      R7,1
         JNE     CLBCNT1
         MOV     @D1,R8
         JMP     CALCCA
CLBCNT1  CI      R7,2
         JNE     CLBCNT2
         MOV     @D2,R8
         JMP     CALCCS
CLBCNT2  CI      R7,3
         JNE     CLBCNT3
         MOV     @D3,R8
         JMP     CALCCA
CLBCNT3  CI      R7,4
         JNE     CLBCNT4
         MOV     @D4,R8
         JMP     CALCCS
CLBCNT4  CI      R7,5
         JNE     CLBCNT5
         MOV     @D5,R8
         JMP     CALCCA
CLBCNT5  CI      R7,6
         JNE     CALCC
         MOV     @D6,R8
CALCCS   S       R8,R1
         JMP     CALCC
CALCCA   A       RI,R1
CALCC    CI      R5,63          ; SEE IF TOO LOW
         JLT     ZERO1          ; IF SO, PRV=0
         CI      R1,1000        ; 1000W MAX
         JLT     CON1
         LI      R1,1000
CON1     MPY     R3,R1          ; PX*4095->R1,R2
         DIV     R5,R1          ; THAT / VLBX -> R1,
                                  REM IN R2, DISC REM
         MPY     R3,R1          ; THAT*4095->R1,R2
         DIV     R4,R1          ; FINAL ANS IN R1, REM
                                  IN R2, DISC REM
         JMP     FCALC          ; SKIP OVER ZERO
ZERO1    CLR     R1
FCALC    B       *R11
OUTPRV   LI      R1,2048        ; MAX OUTPUT
         C       @PRV1,@APRV1   ; COMPARE CALCULATED
                                  WITH ACTUAL
         JEQ     RV2            ; EQUAL DO NOTHING
         JGT     INCP1          ; TO INCREMENT
         C       @APRV1,@DE     ; DONT DECREMENT IF =<
                                  DE
         JLE     RV2
         S       @DE,@APRV1
         JMP     RV2
INCP1    A       @DE,@APRV1
```

```
         C       @APRV1,R1          ; SEE IF TOO BIG
         JLT RV2
         MOV     R1,@APRV1
RV2      C       @PRV2,@APRV2       ; ETC 3 TIMES
         JEQ     RV3
         JGT     INCP2
         C       @APRV2,@DE
         JLE     RV3
         S       @DE,@APRV2
         JMP     RV3
INCP2    A       @DE,@APRV2
         C       @APRV2,R1
         JLT     RV3
         MOV     R1,@APRV2
RV3      C       @PRV3,@APRV3
         JEQ     OUTP
         JGT     INCP3
         C       @APRV3,@DE
         JLE     OUTP
         S       @DE,@APRV3
         JMP     OUTP
INCP3    A       @DE,@APRV3
         C       @APRV3,R1
         JLT     OUTP
         MOV     R1,@APRV3
;
; IF STATUS OFF (EXT SWITCH TO GND)
; THEN ALL OUTPUTS TO 0
;
OUTP     LI      R12,DA1            ; BASE FO DIGITAL INPUT
         TB      3                  ; CRU ADDR OF SW
         JEQ     COUTP              ; HIGH, CONTINUE
         CLR     @APRV1             ; CLR ALL APRVS
         CLR     @APRV2
         CLR     @APRV3
         SETO    R1                 ; ALL 1S
         LI      R12,DA1+(DALSB*2)  ; BASE OF FIRST PRV,
                                      ETC 3 TIMES
         LDCR    R1,12
         LI      R12,DA2+(DALSB*2)
         LDCR    R1,12
         LI      R12,DA3+(DALSB*2)
         LDCR    R1,12
         JMP     CKVBS              ; TO CHECK BUS VOLTS
COUTP    MOV     @APRV1,R1          ; OUTPUT APRVS
         INV     R1                 ; D/A INVERTING TYPE
         LI      R12,DA1+(DALSB*2)  ; BASE OF PRV1
         LDCR    R1,12
         MOV     @APRV2,R1
         INV     R1
         LI      R12,DA2+(DALSB*2)
         LDCR    R1,12
         MOV     @APRV3,R1
         INV     R1
```

```
          LI        R12,DA3+(DALSB*2)
          LDCR      R1,12
;
; IF ANY TWO DIODE BUSSES > H1 THEN OPEN RELAYS
; (ADD DIODES).
; IF ANY TWO DIODE BUSSES < L1 THEN CLOSE RELAYS.
;
CKVBS     LI        R12,DA2+BVH*2
          CLR       R1
          CLR       R2
          C         @VLB1,@H1
          JLT       CKB2H
          INC       R1
CKB2H     C         @VLB2,@H1
          JLT       CKB3H
          INC       R1
CKB3H     C         @VLB3,@H1
          JLT       CKR1H
          INC       R1
CKR1H     CI        R1,2
          JLT       CKB1L
          LDCR      R2,3
          LI        R12,DA3+BVH*2
          LDCR      R2,3
CKB1L     LI        R12,DA2+BVH*2
          CLR       R1
          SETO      R2
          C         @VLB1,@L1
          JGT       CKB2L
          INC       R1
CKB2L     C         @VLB2,@L1
          JGT       CKB3L
          INC       R1
CKB3L     C         @VLB3,@L1
          JGT       CKR1L
          INC       R1
CKR1L     CI        R1,2
          JLT       DOTEMP
          LDCR      R2,3
          LI        R12,DA3+BVH*2
          LDCR      R2,3
;
; AVERAGE 3 EACH THERMISTOR ON EACH PLATE
; (UPPER AND LOWER), IF > UH (OR LH) THEN TURN
: OFF HEATER, IF < UL (OR LL) THEN TURN ON HEATER.
; UH IS THE UPPER PLATE HIGH TEMP, LH IS THE LOWER PLATE HIGH
  TEMP.
; UL IS THE UPPER PLATE LOW TEMP, LL IS THE LOWER PLATE LOW
  TEMP.
;
DOTEMP    LI        R12,DA2
          LI        R3,3
          CLR       R1
          MOV       @T1,R2
```

```
          A         @T2,R2
          A         @T3,R2              ; SUM
          DIV       R3,R1
          MOV       R1,R6
          C         R1,@UH
          JLT       CKLT
          SBZ       BVH+3               ; WAS TOO HIGH, OFF
                                          HEATER
          JMP       CKHB
CKLT      C         R1,@UL
          JGT       CKHB
          SBO       BVH+3               ; WAS TOO LOW, ON
                                          HEATER
CKHB      LI        R12,DA3
          CLR       R1
          MOV       @T4,R2
          A         @T5,R2
          A         @T6,R2
          DIV       R3,R1
          A         R1,R6
          SRL       R6,1
          MOV       R6,@TAVG
          C         R1,@LH
          JLT       CKLB
          SBZ       BVH+3
          JMP       TRRT
CKLB      C         R1,@LL
          JGT       TRRT
          SBO       BVH+3
;
; TRIP BACK ROUTINE
;
; TR IS NO. OF CCC TRIPS (CCCT) THAT OCCUR BEFORE
; ANY POWER SUPPLIES ARE CUT OFF. IF TR = 0 THIS
; ROUTINE IS SKIPPED.
;
; NS IS THE TOTAL NO. OF EQUIVALENT SPAS TAKEN OFF.
; IF THE CCC DISCONNECTS A P.S. THEN THE OTHER P.S. FOR THAT
; BATTERY IS LEFT ALONE. THE ROUTINE FIRST CHECKS IF IT CAN
; REMOVE NS SPAS WITH THE REMAINING 1 SPA P.S.S BEGINNING
; WITH THE 1 SPA P.S. RIGHT AFTER THE BATTERY (OR THE FIRST
; ONE IT SEES) THAT THE CCC CUT OFF. IF IT CAN IT DOES, ELSE
; IT GOES BACK, ADDS THE ONE SPA P.S. AND TAKES OUT THE 2 SPA
; P.S. (ALL THIS IS CHECKED BEFORE ANY ARE ACTUALLY SHUT OFF)
; ONCE THE ROUTINE CUTS OFF THE P.S.(S), IT DOESNT CHECK
; ANYMORE.
;
; POWER SUPPLIES CAN BE COMMANDED OFF BY SETTING THE
; PARAMETER "PS" IN BINARY (YOU FIGURE THE HEX INPUT TO THE
  TTY)
; ANY "1" IN THE WORD MEANS "OFF".
; IN CASE YOU MESS UP AND TURN OFF TOO MANY IN THE ROUTINE,
; P.S.S CAN BE TURNED BACK ON THIS DAY ONLY BY SETTING THE
; PARAMETER "ES". ANY "ONE" MEANS ON.
```

```
; THE FORMAT IS ES,B6K2,B6K1,B5K2,B5K1,...,B2K2,B2K1,B1K2,
  B1K1.(13 BITS)
; WHERE ES IS THE EXTRA 2 SPAS.
          MACRO     AND
          MOV       @'1',R7
          INV       R7
          MOV       @'2'(R2),R8
          INV       R8
          SOC       R7,R8
          INV       R8
          ENDM
;
TRRT      LI        R12,DA1
          TB        4
          JEQ       TODAY               ; 0 IF NIGHT, 1 IF DAY
          SETO      @PSOUT              ; ALL PSS OFF
          CLR       @ES                 ; RESET THOSE THAT WERE
                                          TURNED ON
          CLR       @STT                ; JUST IN CASE
          CLR       @TRF                ; RESET TRIP FLAG IF
                                          NIGHT
          CLR       @DNF                ; DAY NIGHT FLAG TO 0
          SBZ       TFLAG               ; TRIP FLAG ALWAYS 0 IN
                                          NIGHT
BACK      B         OUTPS               ; TO OUTPUT
TODAY     MOV       @DNF,@DNF           ; CHECK FLAG
          JNE       TETR                ; WAS SET, JUMP
          INC       @DNF                ; FIRST DAY CYCLE
          CLR       @CCCT               ; CCC TRIPS
          LI        R9,0FFFH            ; MASK
          MOV       R9,@SLTN
          CLR       @TRF                ; TRIP BACK FLAG
          CLR       @STT
          CLR       @SPOFF
          CLR       @PSOUT
          JMP       BACK                ; FIRST TIME THROUGH GO
                                          TO OUTPUT
TETR      MOV       @TR,@TR             ; IF 0 GET OUT
          JEQ       BACK
          MOV       @TRF,@TRF           ; IF 1 GET OUT
          JEQ       MAJR
          MOV       @NEWRF,R0           ; GO TO NEW ROUTINE?
          JEQ       BACK
          B         NEWR
;
; MAJOR ROUTINE
;
MAJR      LI        R12,CCCIN+(CCCLSB*2)
          STCR      @STT,12             ; INPUT FROM CCC
          CLR       R2
          AND       STT,SLTN            ; IF 0, NOTHING CHANGED
          MOV       R8,@STTSLTN         ; SAVE IT, IF 0 NOTHING
                                          CHANGED
          JEQ       STEP6               ; JUMP AHEAD IN CASE TR
```

```
                                          CHANGED
        ANDI    R7,0FFH           ; STRIP LEADING 0S
        MOV     R7,@SLTN          ; SAVE INVERTED STT
        CLR     R2
CSPA    AND     STTSLTN,MB1       ; SEE IF TRIP ON EACH
                                    BATTERY
        JEQ     NKTRB             ; NO, JUMP
        AND     STTSLTN,B1S1      ; SEE HOW MANY WENT
        JNE     SPO1              ; IF 0, 2 WENT
        INC     @SPOFF            ; IF NOT ZERO, 1 WENT
SPO1    INC     @SPOFF
        INC     @CCCT             : COUNT TRIPS
NKTRB   INCT R2
        CI      R2,12
        JNE     CSPA
STEP6   C       @TR,@CCCT         ; IF TR > CCCT, NOT YET
        JGT     OUTPSB
        INC     @TRF              ; TIME TO DO IT
        CLR     @LBCNT
        CLR     @EPCC
        CLR     @EPC
        LI      R12,DA1
        SBO     TFLAG             ; SIGNAL TRIPBACK
        MOV     @STT,@PSOUT
        MOV     @NEWRF,R0         ; NEW ROUTINE?
        JEQ     STEP7
        MOV     @ET,@ETC
        LI      R0,ETCCI
        MOV     R0,@ETCC
        B       NEWR
STEP7   LI      R1,5
        CLR     R2
STEP8   AND     STT,MB1           ; FIND ONE THAT CCC BID
        JNE     STEP9A
        INCT    R2                ; LOOK AGAIN 4 MORE
                                    TIMES OR...
        DEC     R1
        JNE     STEP8
        CLR     R2                ; ONLY HERE IF B6
        JMP     STEP9
STEP9A  INCT    R2
STEP9   LI      R1,5
        MOV     R2,R0             ; SEE IF CAN TAKE OFF
                                    1SPA PS
STEP10B C       @NS,@SPOFF
        JEQ     OUTPSB
        JLT     OUTPSB
        AND     PSOUT,MB1
        JNE     STEP10A           ; IF 0, CAN DO
        SOC     @B1S1(R2),@PSOUT  ; OR INTO PSOUT WORD
        INC     @SPOFF            ; INC NO. OF SPA OFF
STEP10A DEC     R1                ; SIX BATTS?
        JEQ     STEP11A
        INCT    R2
```

```
          CI        R2,I2
          JNE       STEP10B
          CLR       R2
          JMP       STEP10B
STEP11A   MOV       R0,R2                ; IF HERE, HAVE TAKEN 1
                                           OFF ALL
          LI        R1,5                 ; BATTS AND NOT ENOUGH
                                           YET
STEP11B   C         @NS,@SPOFF
          JEQ       OUTPSB
          JLT       OUTPSB
STEP12    AND       STT,MB1              ; SEE IF CCC DID IT, IF
                                           NOT, DO IT
          JNE       STEP12B
          SZC       @B1S1(R2),@PSOUT     ; PUT SINGLE SPA BACK
          SOC       @B1S2(R2),@PSOUT     ; TAKE OUT DOUBLE SPA
          INC       @SPOFF
STEP12B   DEC       R1
          JEQ       OUTPSB               ; DONE 6 YET?
          INCT      R2
          CI        R2,12
          JNE       STEP11B
          CLR       R2
          JMP       STEP11B
;
OUTPSB    B         OUTPS                ; GET WITHIN RANGE
;
; NEW ROUTINE
; TAKE OFF 1 SPA EVERY X SECONDS (DETERMINED BY SETTING
; THE CAP ET) BEGINNING WITH BATTERY 1 AND GOING THROUGH
; ALL 6 UNTIL HAVE TAKEN OFF NS SPAS.  THIS DOES NOT REMOVE
; ANY SPAS FROM A BATTERY WHOSE CCC TRIPPED OFF A POWER
  SUPPLY.
;
NEWR      DEC       @ETCC
          JNE       OUTPS
          DEC       @ETC
          JEQ       NEWRL1
          LI        R0,ETCCI
          MOV       R0,@ETCC
          JMP       OUTPS
NEWRL1    CLR       R2                   ; TIME TO DO SOMETHING
          MOV       @ETC,@ETC
          LI        R0,ETCCI             ; GETTING READY FOR
                                           NEXT TIME
          MOV       R0,@ETCC
NEWRL2    C         @NS,@SPOFF
          JEQ       OUTPS
          JLT       OUTPS
          AND       STT,MB1              ; IF 1, GO TO NEXT BATT
          JEQ       NEWRL3               ; IF 0, CHECK PSOUT FOR
                                           1 OR 2 SPAS
          INCT      R2
          CI        R2,12
```

```
          JEQ       OUTPS
          JMP       NEWRL2
NEWRL3    AND       PSOUT,B1S2          ; IF NOT 0, 2 SPA IS
                                          ALREADY OUT
          JNE       NEWRL5              ; GO TO NEXT BATTERY
          AND       PSOUT,B1S1          ; IF 0, TAKE OUT 1 SPA
          JNE       NEWRL4              ; AND GET OUT
          SOC       @B1S1(R2),@PSOUT
          INC       @SPOFF
          JMP       OUTPS
NEWRL4    SZC       @B1S1(R2),@PSOUT    ; PUT 1 SPA BACK
          SOC       @B1S2(R2),@PSOUT    ; REMOVE 2 SPA
          INC       @SPOFF
          JMP       OUTPS
NEWRL5    INCT      R2                  ; GO TO NEXT BATTERY
          CI        R2,12
          JNE       NEWRL2
;
; TIME TO OUTPUT
; DONT TURN OFF THOSE THAT CCC TURNED OFF (STT)
; TURN OFF THOSE THAT ARE FLAGGED BY "PS"
; TURN ON THOSE THAN ARE FLAGGED BY "ES" (RESET EACH ORBIT)
;
OUTPS     MOV       @PSOUT,R1
          SZC       @STT,R1             ; DONT TURN OFF IF CCC
                                          OPENED
          SOC       @PS,R1              ; TURN OFF THOSE
                                          FLAGGED BY "PS"
          SZC       @ES,R1              ; TURN ON THOSE
                                          FLAGGED BY "ES"
          LI        R12,POUT+(POUTLSB*2)
          LDCR      R1,13
;
; TIME DELAY
;
TDLP      MOV       @TDLY,RO
TDLPL     DEC       RO
          JNE       TDLPL
;
; PULSE ROUTINE
; SHOULD PUT OUT PULSE EVERY SECOND SO NOT TO GET TIMEOUT
; ON ST HARDWARE. I.E., IF CC QUITS, ST TEST SHUTS DOWN.
; A 60 US PULSE IS PUT OUT EVERY "PL" SOFTWARE CYCLES SO "TD"
; (TIME DELAY BETWEEN CYCLES) WILL HAVE AN EFFECT ALSO.
; THIS ALSO SENSES THE 15V SUPPLY AND STOPS PULSES IF IT
; FAILS.  ALSO, IF THE AVERAGE OF THE SIX PLATE TEMPS
; EXCEEDS "TS", IT STOPS PULSES (CHAMBER FAIL HIGH TEMP)
; THIS ASSUMES THAT THE BATTERY HEATERS CAN KEEP THEM WARM
; IF THE CHAMBER FAILS LOW.
;
          LI        R12,DA1
          TB        V15
          JNE       OUTSAM
          C         @TAVG,@TS
```

```
            JGT         OUTSAM
            DEC         @PLC
            JNE         OUTSAM
            MOV         @PL,@PLC
            LI          R1,10
            SBO         PULSE
TOL         DEC         R1
            JNE         TOL
            SBZ         PULSE
;
; END OF LOOP ..... SO DO FOREVER
;
OUTSAM      B           SAMPLE
MB1         WORD        11B,1100B,110000B,11000000B,1100000000B,
                        110000000000B
B1S2        WORD        01B,0100B,010000B,01000000B,0100000000B,
                        010000000000B
B1S1        WORD        10B,1000B,100000B,10000000B,1000000000B,
                        100000000000B
ZZZ         EQU         $
;
;
            SECTION     TASKRWM,ABSOLUTE
;
            ORG         RWM
;
WP1         BLOCK       32
WP2         BLOCK       32
;
;
CHL0        BLOCK       2                   ; C0 EXAMINE
CHL1        BLOCK       2                   ; C1
CHL2        BLOCK       2                   ; ETC
CHL3        BLOCK       2
CHL4        BLOCK       2
CHL5        BLOCK       2
CHL6        BLOCK       2
CHL7        BLOCK       2
CHL8        BLOCK       2
CHL9        BLOCK       2
CHL10       BLOCK       2                   ; CA
CHL11       BLOCK       2
CHL12       BLOCK       2
CHL13       BLOCK       2
CHL14       BLOCK       2
CHL15       BLOCK       2
PRV1        BLOCK       2                   ; R1
PRV2        BLOCK       2                   ; R2
PRV3        BLOCK       2                   ; R3
APRV1       BLOCK       2                   ; A1  EXAMINE AND SET
                                              THE REST
APRV2       BLOCK       2                   ; A2
APRV3       BLOCK       2                   ; A3
H1          BLOCK       2
```

```
L1          BLOCK     2
H2          BLOCK     2
L2          BLOCK     2
H3          BLOCK     2
L3          BLOCK     2
H4          BLOCK     2
L4          BLOCK     2
H5          BLOCK     2
L5          BLOCK     2
H6          BLOCK     2
L6          BLOCK     2
UH          BLOCK     2
UL          BLOCK     2
LH          BLOCK     2
LL          BLOCK     2
P1          BLOCK     2
P2          BLOCK     2
P3          BLOCK     2
DE          BLOCK     2
TDLY        BLOCK     2                    ; TD
M1          BLOCK     2
M2          BLOCK     2
M3          BLOCK     2
I1          BLOCK     2
I2          BLOCK     2
I3          BLOCK     2
PSOUT       BLOCK     2
PS          BLOCK     2
ES          BLOCK     2
TR          BLOCK     2
NS          BLOCK     2
TRF         BLOCK     2
DNF         BLOCK     2
KI          BLOCK     2
CCCT        BLOCK     2
STT         BLOCK     2
SLTN        BLOCK     2
STTSLTN     BLOCK     2
SPOFF       BLOCK     2
TS          BLOCK     2
PL          BLOCK     2
PLC         BLOCK     2
TAVG        BLOCK     2
ET          BLOCK     2
ETC         BLOCK     2
ETCC        BLOCK     2
NEWRF       BLOCK     2
PR          BLOCK     2                    ; FLAG TO DO POWER STEP
                                             ROUTINE
EPC         BLOCK     2
EPCC        BLOCK     2
LBCNT       BLOCK     2                    ; WHICH STEP
EP          BLOCK     2                    ; COUNTER INITIAL VALUE
```

```
D1          BLOCK     2                ; FIRST POWER DELTA
                                         (ADD)
D2          BLOCK     2                ; SECOND POWER DELTA
                                         (SUB)
D3          BLOCK     2                ; THIRD..ETC...
D4          BLOCK     2
D5          BLOCK     2
D6          BLOCK     2
;
;
VLB1        EQU       CHL1
VLB2        EQU       CHL2
VLB3        EQU       CHL3
VB1         EQU       CHL4
VB2         EQU       CHL5
VB3         EQU       CHL6
VB4         EQU       CHL7
VB5         EQU       CHL8
VB6         EQU       CHL9
T1          EQU       CHL10
T2          EQU       CHL11
T3          EQU       CHL12
T4          EQU       CHL13
T5          EQU       CHL14
T6          EQU       CHL15
;
;
            END       START
```

APPROVAL

HUBBLE SPACE TELESCOPE SIX-BATTERY TEST BED

By J.A. Pajak, J.R. Bush, Jr., and J.R. Lanier, Jr.

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

Joseph L. Randall

JOSEPH L. RANDALL
Director, Information and Electronic Systems Laboratory

☆ U.S. GOVERNMENT PRINTING OFFICE 1990—731—061/20186