MCR-79-1310

MASTER

10-MW_e COLLECTOR SUBSYSTEM SOFTWARE/FIRMWARE FUNCTIONAL REQUIREMENTS SPECIFICATION

6 FEBRUARY 1979

AC03-79ET21007

Prepared by: Paul S. Thames
Paul S. Thames
HAC Lead Engineer

Concurrence: M A Dial
Mike Dials
HFC/HC Lead Engineer

Approved by: W L Hart
Walt Hart
Collector Subsystem Controls Manager

Approved by: P R Brown
P. R. Brown
Program Manager
Collector Systems

see letter dtd Feb 22, 1979 ET21007

CONTENTS

CONTENTS (continued)

TABLES

FIGURES

Acronyms - The following is a list of the acronyms used in this document;

ASCII - American Standard Code for Information Interchange

AZ - Azimuth

BCS - Beam Characterization Subsystem

CLLP - Corridor Lower Limit Point

CMD - Command Processor Module (software module)

CPU - Central Processor Unit

CULP - Corridor Upper Limit Point

DOE - Department of Energy

EL - Elevation

FCP - Field Communications Processor (software module)

HAC - Heliostat Array Controller

HC - Heliostat Controller

HFC - Heliostat Field Control

INIT - Initialization Task

MAX IV - Modcomp Operating System

MMI - Man Machine Interface (software module)

SUN - Sun Vector Module (software module)

1.0 INTRODUCTION

1.1 Purpose

This Functional Requirements Specifications for the 10 MWe Solar Thermal Power Plant Collector Subsystem software shall provide:

a. The system functional requirements to be satisfied and shall be the basis for mutual understanding between the customer and the developer, and

b. The basis for development of software/firmware testing during the System Level Test.

1.2 Functional Summary

The software for the 10 MWe Solar Thermal Power Plant Collector Subsystem (Phase I) shall have the capability to allow operator control of two heliostats, for purposes of testing the heliostats at the Central Receiver Test Facility. This function includes the capability of operator commanded mode control, status display and alarm generation.

2.0 APPLICABLE DOCUMENTS

(1) DE-AC03-79ET21007 Department of Energy Collector Subsystem for the 10 MWe Solar Thermal Central Receiver Pilot Plant Contract Award.

Exhibit I - Statement of Work

Exhibit III - Technical Specification for the Collector Subsystem

(2) Martin Marietta Engineering Practices Manuals G13.1 and G13.2.

(3) 10 MWe Collector Subsystem Software/Firmware Development Plan.

3.0 REQUIREMENTS

3.1 Software Requirements

The software for the 10 MWe Solar Thermal Pilot Power Plant Collector Subsystem (Phase I) shall execute within the Heliostat Array Controller (HAC), Heliostat Field Controller (HFC), and Heliostat Controller (HC) computers to meet the following requirements;

a. Control of the two test heliostats in all modes required to test the heliostats (these modes being a subset of the Phase II operational modes),

b. Monitoring of heliostat modes,

c. Monitoring of heliostat gimbal axis positions,

d. Failure alarm, identification, and response as required for safety, and

e. Safe beam control.

In order to meet these (Phase I) requirements, the software in the HAC shall meet the following requirements;

a. Provide time base,

b. Initiate mode commands to the HFCs, and

c. Address commands to HFC groups or individual HCs.

The firmware in the HFCs shall;

a. Transmit position requirements to the HCs,

b. Transmit status and data to the HAC,

c. Initiate safe stowage commands to the HCs upon loss of HAC communication, and

d. Control groups of HCs.

The firmware in the HCs shall;

a. Determine heliostat azimuth and elevation position requirements,

b. Control gimbal drive motors,

c. Check gimbal axis sensors, and

d. Provide gimbal axis position data to the HAC.

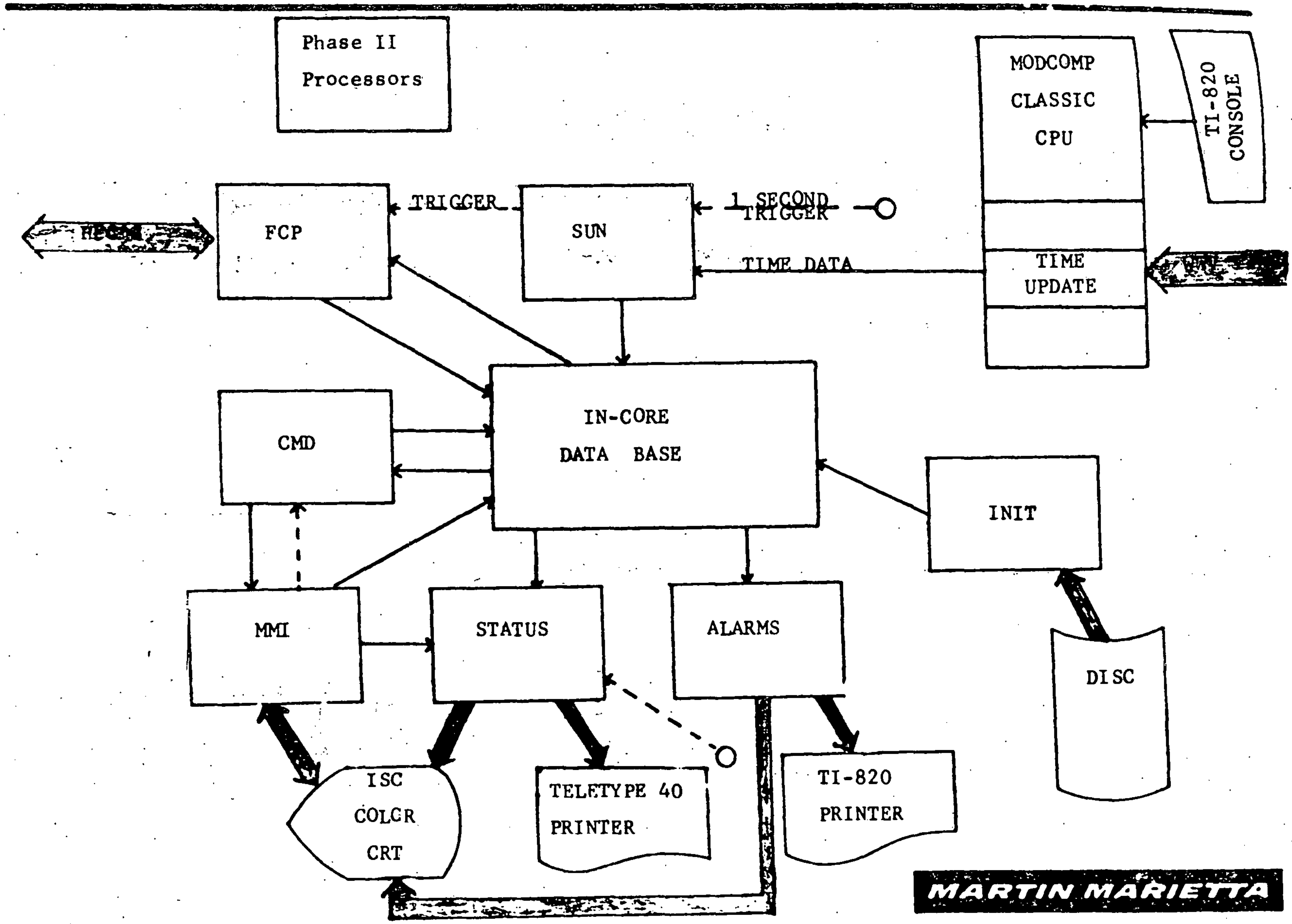
The Collector Subsystem software shall be designed to report any subsystem malfunctions at the HAC console and provide fault isolation information on critical components.

3.2 Associated Derived Software Requirements

In order to meet the software requirements listed above, a more detailed set of requirements have been derived. These derived requirements are a direct consequence of the software requirements from the Statement of Work and Collector Subsystem Technical Description.

3.2.1 HAC Software Derived Requirements - In order to meet the requirements for the software executing in the HAC computer, these requirements have been functionally decomposed into sets or modules. The modules, and their inter-relationships, are depicted in Figure I. The requirements for each module are given in the following paragraphs.

Figure 1 HAC Software Functional Flow Diagram (PHASE 1)



3.2.1.1 Man-Machine Interface Module - The purpose of this module is to accept operator-entered commands, and translate them for use in the computer. The requirements on this module are:

a. Read all operator entered input,

b. Check syntax for valid operator commands,

c. Issue error messages to the HAC console conversational screen partition,

d. Pass valid status commands to the Status Display Processor Module, and

e. Pass valid operational commands to the Command Processor Module.

The list of valid operator commands, and their effect on the heliostats, for Phase I is given in Table 1.

In order to reduce the memory requirements for this program, a root program will be memory resident and will call in disc-resident over-lay programs to process the input commands.

3.2.1.2 Command Processor Module - The purpose of this module is to translate the operator-entered operational commands into series of heliostat commands, controlling heliostat operations. The requirements on this module are;

a. Check command arguments for reasonableness,

b. Check heliostats state for ability to execute commands,

c. Interpret operational commands into individual heliostat commands, and

d. Maintain commanded heliostat mode table.

The list of heliostat command formats are given in Tables 2 through 6, and Tables 11 through 13.

Due to the requirement that each heliostat be independently targetable, and that external priority lists be imposed, the module will require several disc files to maintain priority and target lists, and will require multiple disc accesses to process its commands.

COMMAND	PRIME ARGUMENT	ALTERNATE ARGUMENT	RESULT
Standby	NNNN	#HFC, HC	Stow CLLP CULP
Track	NNNN	#HFC, HC	CULP Target
Stow	NNNN	#HFC, HC	CULP CLLP Stow
Increase	NNNN	#HFC, HC	CULP Target
Decrease	NNNN	#HFC, HC	Target CULP
Position	#HFC, HC, AZ, EL	-	Position to Commanded AZ, EL
Load	-	-	Download all Microprocessor Data to Field
Offline	#HFC, HC	-	Remove Heliostat from Service
Online	#HFC, HC	-	Restore Heliostat to Service
Status	All		Print Snap-Shot of Field Status
Status	Mod/N		Prints List of Heliostats in Mode
Status	IND/#HFC/HC		Print Mode of Selected Heliostat
HC Mark	#HFC, HC	-	Command Heliostat to find Encoder Mark

Table 1. ALLOWABLE OPERATOR COMMANDS (PHASE I)

MESSAGE LAYOUT

APPLICATION HAC HFC

MESSAGE TYPE HC INITIALIZATION COMMAND

PROGRAMMER K. A. Leniger

DATE 2-5-79

1	HEADER BYTE	1	06_{16}
	#		3 bits HFC#; 5 bits HC#
	CAZ		HC current azimuth
5	CEL	5	HC current elevation
	BAZ		HC bias azimuth
	BEL		HC bias elevation
10		10	
	HX		Heliostat field x-coord. (scaled binary point 14.)
15	HY	15	Heliostat field y-coord. (scaled binary point 14.)
	HZ		Heliostat field z-coord. (scaled binary point 14.)
20	CHECKSUM	20	8 bit checksum such that <u>all</u> message bytes sum to zero.
25		25	
30		30	

Table 2. HC INITIALIZATION COMMAND FORMAT

MESSAGE LAYOUT

APPLICATION HAC HFC

MESSAGE TYPE BEAM POINTING COMMAND

PROGRAMMER K. A. Leniger

DATE 2-5-79

1	HEADER BYTE	08 TRACK RECEIVER; 09 TRACK ALT-1; 0A TRACK ALT-2; 0B TRACK BCS; 0C TRACK CULP; 0D TRACK CLLP
2	#	3 BITS HFC#; 5 BITS o's
3–6	HC SELECT MASK	32 BIT HC SELECT MASK FOR THIS COMMAND; 1 = SELECT THIS HC
7–9	BPX	Beam Pointing Target x-coord. (scaled at binary point 14).
10–12	BPY	Beam Pointing Target y-coord. (scaled at binary point 14).
13–15	BPZ	Beam Pointing Target z-coord. (scaled at binary point 14).
16–19		don't care
20	CHECKSUM	8 bit checksum such that all message bytes sum to zero.
21–32		

Table 3. BEAM POINTING COMMAND FORMAT

MESSAGE LAYOUT

APPLICATION HAC HFC

MESSAGE TYPE CORRIDOR WALK START-UP COMMAND

PROGRAMMER K. A. Leniger

DATE 2-5-79

1	HEADER BYTE	1	11_{16} C.W. UP-A 12_{16} C.W. DN-A 15_{16} C.W. UP-B 16_{16} C.W. DN-B
	#		3 bits HFC# 5 bits o's
	HC SELECT		32 bit HC select mask for this command. 1 = select this HC
5	MASK	5	
10		10	
			don't care
15		15	
20	CHECKSUM	20	8 bit checksum such that <u>all</u> message bytes sum to zero.
25		25	
30		30	

Table 4. CORRIDOR WALK START-UP COMMAND FORMAT

MESSAGE LAYOUT

APPLICATION HAC HFC

MESSAGE TYPE AZ/EL POINTING COMMAND

PROGRAMMER K. A. Leniger

DATE 2-5-79

1	HEADER BYTE	18 AZ/EL STOW 19 AZ/EL SEEK MARKS 1A AZ/EL WASH 1B AZ/EL MAINT.
2	#	3 bits HFC# 5 bits o's
3–6	HC SELECT MASK	32 bit HC select mask for this command. 1 = select this HC
7–8	AZ	Command azimuth
9–10	EL	Command elevation
11–19		don't care
20	CHECKSUM	8 bit checksum such that <u>all</u> bytes of the message sum to zero.
21–32		

Table 5. AZIMUTH/ELEVATION POINTING COMMAND FORMAT

MESSAGE LAYOUT

APPLICATION HAC HFC **MESSAGE TYPE** STATUS POLL

PROGRAMMER K. A. Leniger **DATE** 2-5-79

1	HEADER BYTE	1	40_{16} STATUS POLL 4-HCs 60_{16} STATUS POLL 1-HC.
	#		3 bits HFC#; 5 bits HC# if single HC POLL.
			don't care
	CHECKSUM		all message bytes sum to zero.
5		5	
10		10	
15		15	
20		20	
25		25	
30		30	

Table 6. STATUS POLL COMMAND FORMAT

Table 7. FOUR [illegible]IOSTAT STATUS RESPONSE FORMAT

MESSAGE LAYOUT

APPLICATION HFC HAC

MESSAGE TYPE 4-HC STATUS RESPONSE

PROGRAMMER K. A. Leniger

DATE 2-5-79

1	HEADER BYTE	50_{16}
	#	3 bit HFC#; 5 bit HC# of the first HC in the group
5	COMMAND RECEIVED MASK	32 bits of command received mask 0 = this HC did not receive a command 1 = HC received a command OK
	HC STATUS	See breakdown of HC status - HC status for HC#n
10	CAZ	Current azimuth for HC#n
	CEL	Current elevation for HC#n
	HC STATUS	See breakdown of HC status - HC status for HC# n+1
15	CAZ	Current azimuth for HC# n+1
	CEL	Current elevation for HC# n+1
20	HC STATUS	See breakdown of HC status - HC status for HC# n+2
	CAZ	Current azimuth for HC# n+2
	CEL	Current elevation for HC# n+2
25	HC STATUS	See breakdown of HC status - HC status for HC# n+3
	CAZ	Current azimuth for HC# n+3
30	CEL	Current elevation for HC# n+3
	HFC STATUS	See breakdown of HFC status - HFC status this HFC
	HFC CMD	Current command received by this HFC or 80_{16} if none.
	CHECKSUM	8 bit checksum such that <u>all</u> message bytes sum to zero.

MESSAGE LAYOUT

APPLICATION HFC HAC

MESSAGE TYPE 1-HC STATUS RESPONSE

PROGRAMMER

DATE 2-5-79

1	HEADER BYTE	1	70_{16}
	#		3 bits HFC#; 5 bits HC#/
	HC STATUS		See breakdown of HC status.
5	CAZ	5	Current azimuth this HC.
	CEL		Current elevation this HC.
			don't care
10	CHECKSUM	10	8 bit checksum such that <u>all</u> message bytes sum to zero.
15		15	
20		20	
25		25	
30		30	

Table 8. ONE HELIOSTAT STATUS RESPONSE

MESSAGE LAYOUT

APPLICATION HC STATUS BYTES **MESSAGE TYPE** HC STATUS BIT BREAKDOWN

PROGRAMMER K. A. Leniger **DATE** 2-5-79

BIT POSITION

1	MA	MSB — Mark encountered azimuth
2	ME	Mark encountered elevation
3	PC	Position compare within $\pm$ TBD bits in both az and el
4	MANC	Mark encountered az and no compare
5	MENC	Mark encountered el and no compare
6	CE	HFC detected comm. error
7	NM	Az or El gimbal not moving when commanded.
8	EE	Az or El encoder jumped too much
9	DIR	Az or El gimbal moving in the wrong direction.
10–11	MODE	see MODE table
12–14	SUBMODE	see SUB MODE table
15–16		don't care (16 = LSB)
17–32		

MODE	SUB MODE	
00	-	Restart
	000	Power-on
	110	Init
	111	Comm. Error
01	-	Beam Pointing
	000	Track Receiver
	001	Track Alt-1
	010	Track Alt-2
	011	Track BCS
	100	Track CULP
	101	Track CLLP
	110	-
	111	-
10	-	Corridor Walking
	001	Up Corridor-A
	010	DN Corridor-B
	101	Up Corridor-B
	110	DN Corridor-B
11	-	Az/El Pointing
	000	Stow
	001	Seek Marks
	010	Wash
	011	Maint.

Table 9. HELIOSTAT CONTROLLER STATUS BIT BREAKDOWN

MESSAGE LAYOUT

APPLICATION HFC STATUS BYTES MESSAGE TYPE HFC STATUS BREAKDOWN

PROGRAMMER K. A. Leniger DATE 2-5-79

BIT POSITION

1	RST	1	MSB → HFC restarted
2	CE		HFC/HAC comm. error
3–4	I PHASE		HFC initialization phase = 0 just restarted; expecting CULPs 1 got CULPs; expecting CLLPs 2 got CLLPs; expecting DELTAs 3 If RST = 1: got DELTAS; expecting corridor assignments RST = 0: normal; any corridor assignment command allowed.
5–12		5, 10	don't care
13–14	WAS		Corridor-A State: 0 idle 1 up 2 down
15–16	WBS	15	LSB → Corridor-B State: 0 idle 1 up 2 down
17–32		20, 25, 30	

Table 10. HFC STATUS BREAKDOWN

MESSAGE LAYOUT

APPLICATION HAC HFC MESSAGE TYPE SUN/SYNC

PROGRAMMER K. A. Leniger DATE 2-5-79

1	HEADER BYTE	80_{16} Means SUN/SYNC.
	SPN	HC number of first of four consecutive HCs to be status polled this second.
5	SX	Sun unit vector x-coordinate (scaled at binary point 4).
		don't care
	SY	Sun unit vector y-coordinate (scaled at binary point 4).
10		don't care
	SZ	Sun unit vector z-coordinate (scaled at binary Point 4).
15		don't care
	CHECKSUM	8-bit Checksum such that all message bytes, including the Checksum, sum to zero.
20		
25		
30		

Table 11. SUN POSITION COMMAND FORMAT

MESSAGE LAYOUT

APPLICATION HAC HFC

MESSAGE TYPE HFC INIT. COMMAND SUBTYPES 0,1,2

PROGRAMMER K. A. Leniger

DATE 2-5-79

1	HEADER BYTE	1	CX_{16} = CULPs; DX_{16} = CLLPs; EX_{16} = DELTAs (X is HFC#)
	XA		Corridor-A X-parameter at (binary point 14).
5	YA	5	Corridor-A Y-parameter (binary point 14).
	ZA		Corridor-A Z-parameter (binary point 14).
10		10	
	XB		Corridor-B X-parameter (binary point 14).
15	YB	15	Corridor B Y-parameter (binary point 14).
	ZB		Corridor-B Z-parameter (binary point 14).
20	CHECKSUM	20	8 bit checksum such that <u>all</u> message bytes sum to zero.
25		25	
30		30	

Table 12. HFC INITIALIZATION COMMAND FORMAT (SUBTYPES 0, 1, 2)

MESSAGE LAYOUT

APPLICATION HAC HFC

MESSAGE TYPE HFC INIT. COMMAND SUBTYPE 3

PROGRAMMER K. A. Leniger

DATE 2-5-79

Byte	Field	Description
1	HEADER BYTE	FX_{16} = Corridor Assignments
2		don't care (X = HFC#)
3–6	WIRE ASSIGNMENT MASK	32 bits of corridor assignments 0 = Corridor-A this HC 1 = Corridor-B this HC
7–19		don't care
20	CHECKSUM	8 bit checksum such that all message bytes sum to zero.
21–32		

Table 13. HFC INITIALIZATION COMMAND FORMAT (SUBTYPE 3)

3.2.1.3 Sun Vector Module - The purpose of this module is to calculate sun position, required by the HC for heliostat position determination. The requirements on this module are;

a. Calculate a unit sun vector as a function of universal time, on a once per second basis.

b. Format the sun position unit vector into the format required for message packets and store in the data base.

The format for the sun vector message packet is shown in Table 11.

3.2.1.4 Field Communications Processor Module - The purpose of this module is to control communications with the HFC computers. The requirements on this module are;

a. Synchronize field operations by transmitting to all HFC computers a sun vector command, once a second,

b. Generate, at the proper time in the one second time frame, the polling commands for each HFC,

c. Receive the HC and HFC status in response to each polling command, and save same in the data base until next update,

d. Transmit to the proper HFCs, the operational commands generated by the Command Processor Module, and

e. Detect communications lines failures, sending same to Alarm Processor Module for display,

f. Mark heliostats off-line and/or marked.

The timing of the one second communications frame is given in Figure 2.

3.2.1.5 Alarm Processor Module - The purpose of this module is to detect and display error conditions reported by the HCs, HFCs, and other software modules. The requirements on this module are:

a. Monitor the heliostat status being returned from the field,

b. Detect error conditions reported by the HC in that status,

c. Report the alarms to the operator, using the alarms printer and a protected area of the HAC console display,

d. Display alarms detected by other software modules,

e. Maintain service/out-of-service status for the heliostats.

The data being returned for each heliostat is shown in Tables 9 and 10.

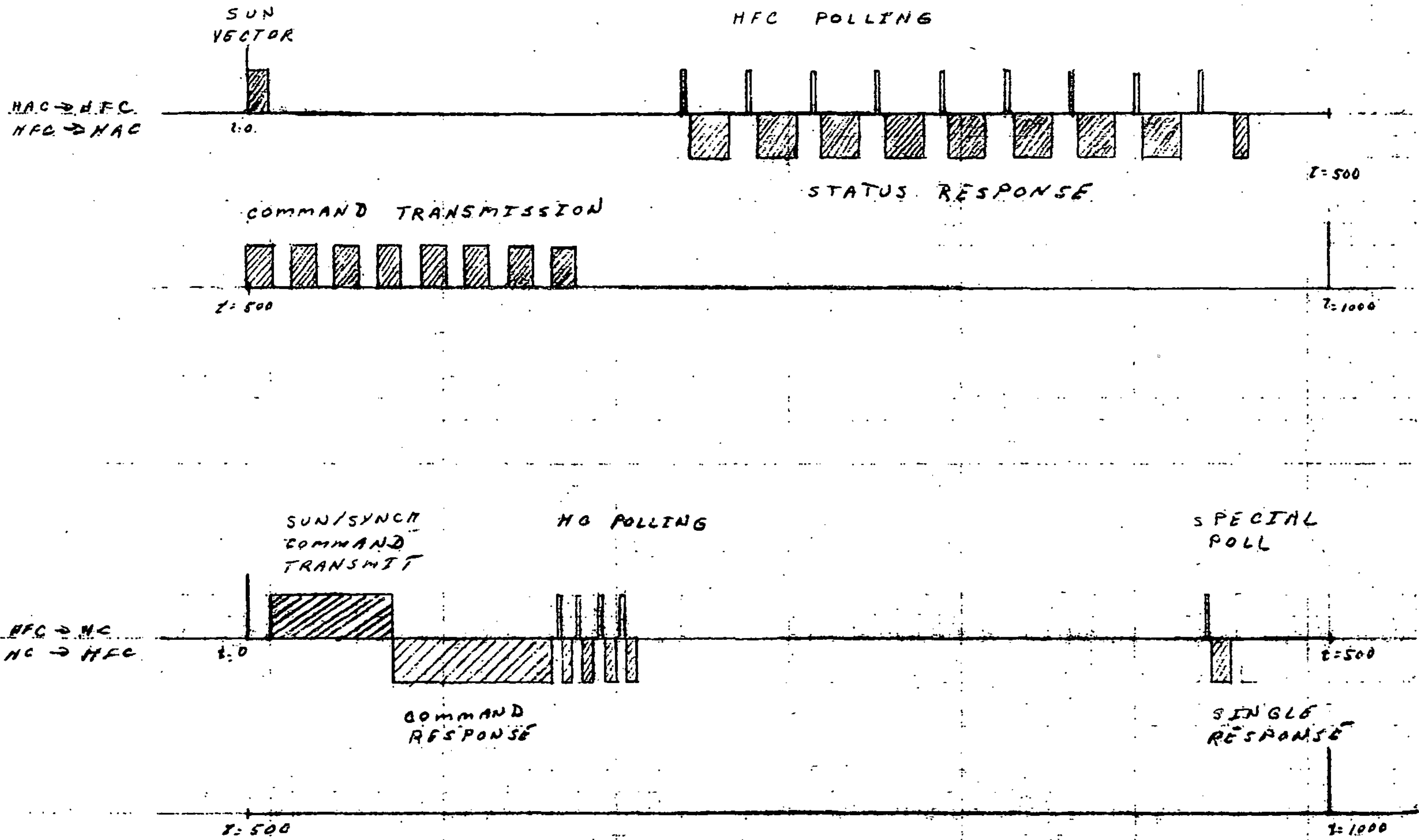


Figure 2. Communications Timing Diagram.

3.2.1.6 Status Display Module - The purpose of this module is to provide the operator with a continuous status of the field, and a more detailed status on demand. The requirements on this module are;

a. Monitor heliostat status, from status data reported by the HCs,

b. Format field status,

c. Display same on protected area of HAC console display, updating display as required,

d. Respond to operator-entered commands for display status, as reported from the MMI module, and format the status requested, and

e. Output the formatted status to the status printer.

NOTE: Continuous and demand status formats are presented in Figure 3.

3.2.1.7 Data Base Module - The purpose of this module is to provide a common data area accessible by all applications tasks, to be used for inter-task communications. The requirements on this module are;

a. Be a global common area, accessible to all applications tasks,

b. Provide an initialization task to initialize the values in the data base from values on the disc, and

c. Provide task start up sequence to the operating system.

3.2.1.8 Operating System Modifications Module - The purpose of this module is to provide the modifications to the MAX IV Operating System which will make it compatible with the MODCOMP CLASSIC computer system in the configuration purchased for this project. The requirements on this module are;

a. Generation of an operating system compatible with the peripheral equipment on the computer system,

b. Addition of the handlers to communicate with the peripherals of the system, with special emphasis on the Universal Communications Chassis (#1930) through which communications with the heliostat field is performed,

c. Universal time maintenance, using the WWVB Clock as reference standard, and

d. Power fail/auto-restart capability.

3.2.2 HFC Firmware Derived Requirements - The derived requirements for the firmware executing in the HFC microprocessor are;

a. Maintain communications with the HAC computer,

HELIOSTAT STATUS: MM/DD/YY HH:MM:SS

ON-RECEIVER				OTHER OPERATIONAL			OUT-OF SERVICE	
TRACK	ALT 1	ALT 2	BCS	STANDBY	STOW	TRANSITION	WASH	BROKEN
#	#	#	#	#	#	#	#	#

FIELD STATUS

HELIOSTAT X: MODE - $\left\{\begin{array}{lll} \text{TRACK} & \text{BCS} & \text{TRANSITION} \\ \text{ALT 1} & \text{STANDBY} & \text{WASH} \\ \text{ALT 2} & \text{STOW} & \text{BROKEN} \end{array}\right\}$ AZIMUTH - XXX ELEVATION - XXX TIME HH:MM:SS

INDIVIDUAL STATUS

MODE: XXXXXXXXXX (TEXT OF SELECTED MODE) TIME HH:MM:SS

HFC #: HC # ..... HFC #: HC # 10 PER LINE, 1 BLANK LINE BETWEEN LINES

MODE STATUS

Figure 3 Status Display Formats

b. Transmit the received commands to all 32 HCs connected on the HFC,

c. Poll the specified four HCs in response to a polling command from the HAC,

d. Transmit the status received from the four HCs to the HAC, upon request from the HAC,

e. Detect communications errors in messages from HAC and HC,

f. On loss of communications with HAC computer, control stow of all 32 heliostats using an approximate corridor walk (using last received sun vector).

3.2.3 HC Firmware Derived Requirements - The derived requirements for the firmware executing in the HC microprocessor are;

a. Maintain communications with the HFC, detecting communications errors,

b. Calculate the desired azimuth and elevation angles based on sun vector and operational commands from the HFC,

c. Control the azimuth and elevation motors to achieve the desired position,

d. Maintain the azimuth and elevation encoder interfaces to provide gimbal angle information,

e. Provide HC status information to the HFC upon a poll command,

f. Execute an automatic re-initialization upon detection of major errors, and

g. Detect major errors of heliostat operation (i.e., rate error detection).

3.3 <u>Controls Subsystem Input/Output Interfaces</u>

This section will define the external interfaces for the software which executes in the HAC computer, and the firmware which executes in the HFC and HC microprocessors.

3.3.1 HAC Software Input/Output Interface Requirements

3.3.1.1 HAC Input Requirements

3.3.1.1.1 Operator Input Commands - These input commands are entered to the HAC computer from the HAC console, and are used to operationally control the heliostat field. The list of commands to which the HAC software will respond is presented as Table I.

3.3.1.1.2 Universal Time - Universal time is required by the HAC software in order to calculate the sun position vector, required by the HC's for beam pointing. Time is entered into the HAC computer from the Tru-Time 60DC WWVB Clock, which is a WWV Time Clock Receiver and Generator. The computer interface is a serial ASCII interface (RS-232C), and time is generated every minute. Second by second updates in the computer are generated by the operating system, with time re-synchronized with each minute input.

3.3.1.1.3 HC Status Data - The HC status data is required by the HAC software for status displays on the HAC console and status printer. Status data is sent to the HAC by the HFC in response to status polling requests. The polling requests are generated such that one eighth of the heliostat field is polled each second, and thus the entire field is statused every eight seconds. The format of the status data for a HC is presented in Tables 7 and 9.

3.3.1.2 HAC Output Requirements

3.3.1.2.1 HFC Commands - The HAC software generates command sequences and transmits them to the field of HFCs, to operationally control the heliostat field, and display the status of the field. These command sequences are generated and transmitted once per second, and thus may take up to 32 seconds to command all HCs. The format of these sequences is given in Tables 2-13. The relative timing of these sequences is displayed as Figure 2.

3.3.1.2.2 Field Status - The status of the heliostat field is calculated and displayed on the HAC console automatically. This status data is maintained in the HAC from the field status information returned from the field, and the display is updated, as required. The format for this status display is shown in Figure 3. In addition, more detailed status information can be requested through the HAC console. This information is then printed on the status printer, and as an operator option, can be displayed on the HAC console. Three formats are given as an operator option. These formats are displayed as Figure 3.

3.3.1.2.3 Heliostat Alarms - Heliostat alarm information is generated by the HAC software when irregular heliostat field operation is detected from the heliostat status information received from the field. This alarm data is displayed on the HAC console and printed on the alarm printer.

3.3.2 HFC Input and Output Requirements

3.3.2.1 HFC Input Requirements

3.3.2.1.1 HFC Input from HAC Computer - The input required by the firmware in the HFC is the once per second command sequence generated by the HAC software (See Tables 2-13). The HFC uses this command sequence to generate the output sequences it sends to the 32 HCs associated with it.

3.3.2.1.2 HFC Input from HC Microprocessor - The input required by the HFC firmware is HC status information received as the result of a status poll command. This status information is used to generate status packets transmitted to the HAC computer. The format of the status packet received from the HC is presented in Table 19.

MESSAGE LAYOUT

APPLICATION HFC HC

MESSAGE TYPE SUN/SYNC/CMD BEAM POINTING

PROGRAMMER K. A. Leniger

DATE 2-7-79

1	HEADER BYTE	28_{16} Receiver 29_{16} Alt-1 $2A_{16}$ Alt-2 $2B_{16}$ BCS $2C_{16}$ CULP (standby) $2D_{16}$ CLLP
	SX	Sun unit vector x-coord. at binary pt. 4.
5	SY	Sun unit vector y-coord. at binary pt. 4.
10	SZ	Sun unit vector z-coord. at binary pt. 4.
	WIRE-A SELECT MASK	32 bit HC select mask of HCs to point at corridor-A.
15	WAX	Corridor-A x-coord. at binary point 14.
20	WAY	Corridor-A y-coord. at binary point 14.
	WAZ	Corridor-A z-coord. at binary point 14.
25	WIRE-B SELECT MASK	32 bit HC select mask of HCs to point at Corridor-B.
30	WBX	Corridor-B x-coord. at binary point 14.
	WBY	Corridor-B y-coord. at binary point 14.

Table 14. HC SUN/SYNCHRONIZATION COMMAND FORMAT (BEAM POINTING)

MESSAGE LAYOUT

APPLICATION HFC HC

MESSAGE TYPE SUN/SYNC/CMD BEAM POINTING (CONTINUED)

PROGRAMMER K. A. Leniger

DATE 2-7-79

(continued)

35	WBZ	Corridor-B z-coord. at binary point 14.
	BEAM POINTING SELECT MASK	32 bit select mask of HCs to beam point.
40		
		don't care
45	BPX	Beam pointing x-coord at binary point 14.
	BPY	Beam pointing y-coord at binary point 14.
50		
	BPZ	Beam pointing z-coord at binary point 14.
	CHECKSUM	8 bit checksum such that <u>all</u> message bytes sum to zero.
55		
60		
65		

Table 14. HC SUN/SYNCHRONIZATION COMMAND FORMAT (BEAM POINTING) continued

MESSAGE LAYOUT

APPLICATION: HFC HC | MESSAGE TYPE: SUN/SYNC/CMD AZ/EL POINTING

PROGRAMMER: K. A. Leniger | DATE: 2-7-79

1	HEADER BYTE	78_{16} Stow; 79_{16} Seek Marks; $7A_{16}$ Wash; $7B_{16}$ Maint.
	SX	Sun unit vector x-coord at binary point 4.
5	SY	Sun unit vector y-coord at binary point 4.
10	SZ	Sun unit vector z-coord at binary point 4.
	WIRE-A SELECT MASK	32 bit HC select mask of HCs to point at corridor-A.
15	WAX	Corridor-A x-coord. at binary point 14.
20	WAY	Corridor-A y-coord. at binary point 14.
	WAZ	Corridor-A z-coord. at binary point 14.
25	WIRE-B SELECT MASK	32 bit HC select mask of HCs to point at Corridor-B.
30	WBX	Corridor-B x-coord at binary point 14.
	WBY	Corridor-B y-coord at binary point 14.

(continued)

Table 15. HC SUN/SYNCHRONIZATION COMMAND FORMAT (AZIMUTH/ELEVATION POINTING)

MESSAGE LAYOUT

APPLICATION HFC HC

MESSAGE TYPE SUN/SYNC/CMD AZ/EL POINTING (CONTINUED)

PROGRAMMER K. A. Leniger

DATE 2-7-79

35	WBZ	Corridor-B z-coord at binary point 14.
	AZ/EL POINTING SELECT MASK	32 bit select mask of HCs to AZ/EL point.
40		
	CAZ	Command azimuth
	CEL	Command elevation
45		don't care
		don't care
50		
		don't care
	CHECKSUM	8 bit checksum such that all message bytes sum to zero.
55		
60		
65		

Table 15. HC SUN/SYNCHRONIZATION COMMAND FORMAT (AZIMUTH/ELEVATION POINTING) continued

MESSAGE LAYOUT

APPLICATION HFC HC

MESSAGE TYPE SUN/SYNC/CMD HC INITIALIZATION

PROGRAMMER K. A. Leniger

DATE 2-7-79

1	HEADER BYTE	3 bits = zero 5 bits HC#
	SX	Sun unit vector x-coord at binary point 4.
5	SY	Sun unit vector y-coord at binary point 4.
10	SZ	Sun unit vector z-coord at binary point 4.
	WIRE-A SELECT MASK	32 bit HC select mask of HCs to point at Corridor-A.
15	WAX	Corridor-A x-coord at binary point 14.
20	WAY	Corridor-A y coord at binary point 14.
	WAZ	Corridor-A z-coord at binary point 14.
25	WIRE-B SELECT MASK	32 bit HC select mask of HCs to point at Corridor-B.
30	WBX	Corridor-B x-coord at binary point 14.
	WBY	Corridor-B y-coord at binary point 14.

(continued)

Table 16. SUN/SYNCHRONIZATION COMMAND FORMAT (HC INITIALIZATION)

MESSAGE LAYOUT

APPLICATION HFC HC

MESSAGE TYPE SUN/SYNC/CMD HC INITIALIZATION (CONTINUED)

PROGRAMMER K. A. Leniger

DATE 2-7-79

(continued)

35	WBZ	35	Corridor-B z-coord at binary point 14.
	CAZ		Current azimuth
	CEL		Current elevation
40		40	
	BAZ		Bias azimuth
	BEL		Bias elevation
45		45	
	HX		Heliostat x-coord at binary point 14.
	HY		Heliostat y-coord at binary point 14.
50		50	
	HZ		Heliostat z-coord at binary point 14.
	CHECKSUM		8 bit checksum such that all message bytes sum to zero.
55		55	
60		60	
65		65	

Table 16. HC SUN/SYNCHRONIZATION COMMAND FORMAT (HC INITIALIZATION) continued

MESSAGE LAYOUT

APPLICATION HC HFC MESSAGE TYPE COMMAND RESPONSE

PROGRAMMER K. A. Leniger DATE 2-7-79

1	C. R. BYTE	1	3 bits = 6_{16} 5 bits HC#
5		5	
10		10	
15		15	
20		20	
25		25	
30		30	

Table 17. HC COMMAND RESPONSE FORMAT

MESSAGE LAYOUT

APPLICATION HFC HC

MESSAGE TYPE STATUS POLL

PROGRAMMER K. A. Leniger

DATE 2-7-79

Byte	Field	Byte	Description
1	HEADER BYTE	1	3 bits = 4_{16} 5 bits HC#
	CHECKSUM		2's complement of first byte!

Table 18. HC STATUS POLL COMMAND FORMAT

MESSAGE LAYOUT

APPLICATION HC HFC

MESSAGE TYPE STATUS RESPONSE

PROGRAMMER K. A. Leniger

DATE 2-7-79

1	HEADER BYTE	1	3 Bits = 5_{16}; 5 Bits HC#
	HC STATUS		See HC status bit breakdown for HFC HAC status response
	CAZ		Current azimuth
5		5	
	CEL		Current elevation
	CHECKSUM		8 bit checksum such that <u>all</u> message bytes sum to zero.
10		10	
15		15	
20		20	
25		25	
30		30	

Table 19. HC STATUS RESPONSE FORMAT

3.3.2.2 HFC Output Requirements

3.3.2.2.1 HFC Output to HC - The output generated by the HFC firmware for the HC consists of command sequences. These command sequences consist of sun vector commands, status polling commands, and operational commands. The format of these commands is shown in Tables 14 through 16.

3.3.2.2.2 HFC Output to HAC - The output generated by the HFC firmware for the HAC consists of status information packets. The format of these packets is shown in Tables 7 and 8.

3.3.3 <u>HC Input/Output Requirements</u>

3.3.3.1 HC Input Requirements

3.3.3.1.1 HC Input from HFC - The input required by the HC firmware from the HFC consists of the command sequences. These command sequences consist of sun vector, status polling command, and operational commands. Format of these sequences is shown in Table 14 through 16.

3.3.3.1.2 HC Input from Encoder - The input required by the HC firmware is a four-bit counter value giving magnitude and direction of travel for each of the elevation and azimuth axes.

3.3.3.2 HC Output Requirements

3.3.3.2.1 HC Output to HFC - The output generated by the firmware in the HC for the HFC is the HC status packet. This information is generated and transmitted in response to a status poll command. The format of this status packet is presented in Table 18.

3.3.3.2.2 HC Output to Motor Controller - The outputs generated by the HC firmware are the motor speed command signals (slow and slew speeds) and motor direction commands for azimuth and elevation drive motors.

3.4 <u>Data Base</u>

The data base in the HAC computer will contain data required by more than one software module in the HAC, or data which is variable, but must be initialized to preset values before the software tasks begin execution. Initialization data will reside on a file on the disc in card image format. An initialization task will be executed automatically upon operating system start-up. This task will read the data base initialization file, convert the data and initialize the data base, configure the HAC console for partitioned screen operation, and initiate task execution sequences.

3.5 <u>Limitations and Constraints</u>

The limitations and constraints on the software to execute in the HAC computer and the firmware to execute in the HFC and HC microprocessors is discussed in this section.

3.5.1 HAC Software Limitations and Constraints - The software to execute in the HAC computer for both Phase I and Phase II is constrained to operate in 128K words of memory. A design goal of the Phase I software will be to have a minimum 35% memory margin for Phase II software development. A further constraint on the Phase I software is that communications with the heliostat field shall occur periodically on one second time increments.

3.5.2 HFC Firmware Limitations and Constraints - The firmware in the HFC will execute with the following constraints and limitations;

a. The firmware will execute in 4096 8-bit bytes of read-only memory and 1152 8-bit bytes of random access memory,

b. The firmware will control up to 32 HCs connected to it, communicating over a 9600 baud data line,

c. Initialization of the firmware programs and data will take up to four seconds,

d. The HFC firmware will communicate with the HAC over a 19,200 baud data line,

e. The HFC firmware will not attempt to propogate sun position, in case of HAC communication failure,

f. The HFC firmware will control its HCs during the corridor walk; however, no more than two corridors may be assigned to the HCs on the HFC, and each corridor must be linear and single segment, and

g) The firmware will process status data for no more than four HCs in a one second time frame. Therefore, it will take eight seconds to return status data from all HCs on the HFC.

3.5.3 HC Firmware Limitations and Constraints - The firmware in the HC will execute with the following constraints;

a. The firmware will execute in 4096 8-bit bytes of read-only memory and 256 8-bit bytes of random access memory,

b. The firmware will communicate with the HFC over a 9600 baud data rate line.