Rotation Utility Module Specification

James Francis

MIT 64-30110.2501

Rev. C

May 2, 1994

Contents

Figures

1 Introduction

1.1 Scope

This document describes the components that make up the Rotation Utility Software Module (ROTUTIL). This module consists of a set of functions that the Flight Software uses to manage ASM rotations.

This document does not describe the overall design of the EDS, nor does it describe the operating details of each component in this module. It assumes that the reader is extremely familiar with the Flight Software Global Design, with the "C" programming language and with the architecture of the EDS.

This document should be of primary interest to Flight Software Engineering, Flight Hardware Engineering, Ground Software Engineering and Software Quality Assurance.

1.2 References

MIT 64-02012 Rev. D .... "EDS-ASM Science Requirements"
MIT 64-03006 Rev. E....................... "ASM/EDS Interface Control Document"
MIT 64-02006.01 Rev. E.................. "XTE EDS Hardware Functional Description and Requirements"
MIT 64-30102 Rev. F....................... "Flight Software Global Design"
MIT 64-30110.0901 Rev. D .... "Front Registers Module Specification" (FREG)
MIT 64-30110.2001 Rev. D .... "Utility Module Specification" (UTIL)

1.3 Notations

This document assumes the following notations:

UINT16... Refers to an unsigned 16-bit value.
BOOL .... Refers to a 16-bit value which represents a TRUE or FALSE condition.

1.4 Module Perspective

The following diagram is a copy of the Level 1 data flow diagram from the **Flight Software Global Design**, Section 4.1.2. The Rotation Utility Module does not appear directly in this data flow, but has a scope which affects items indicated by the shaded areas:

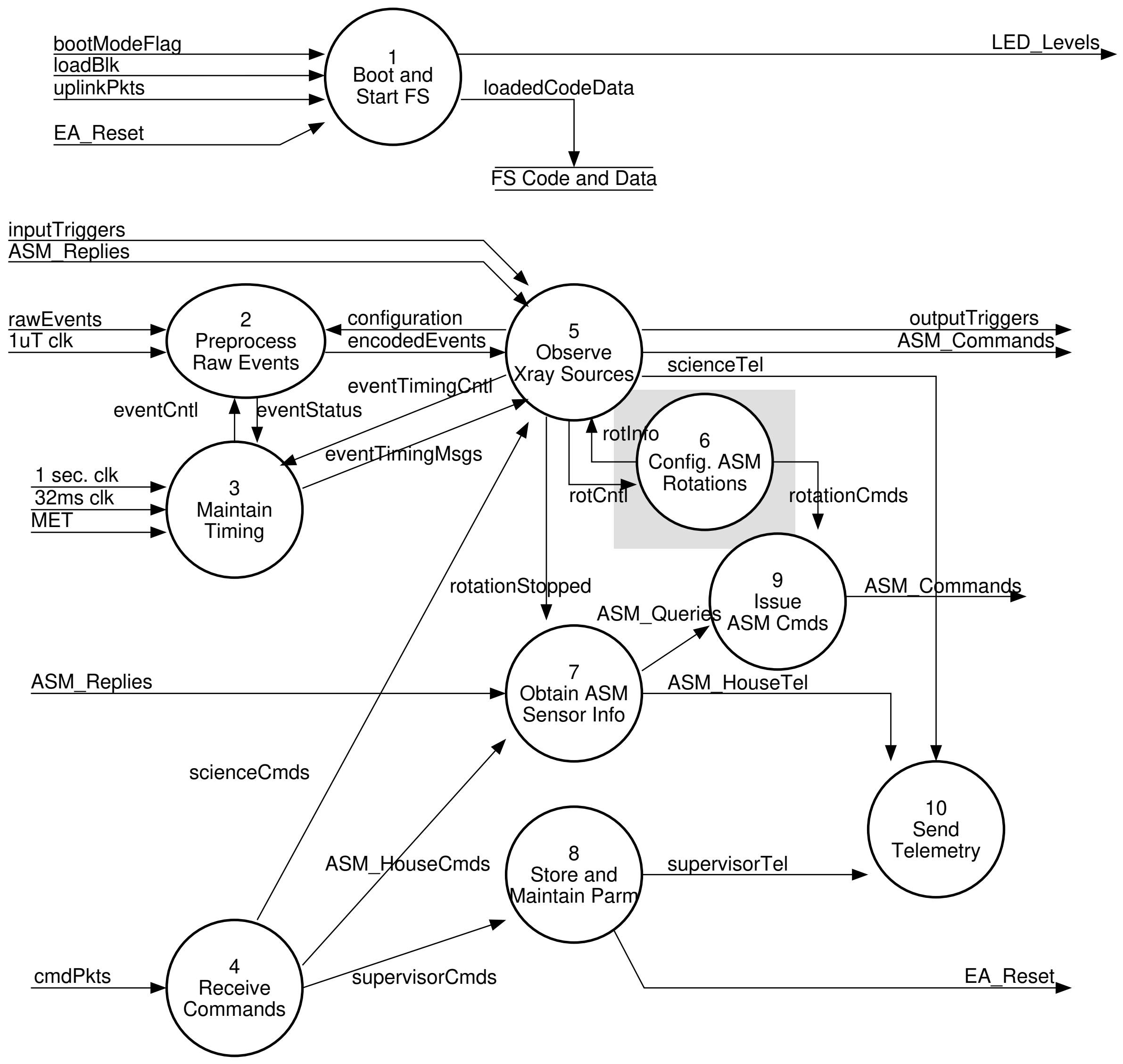

Figure 1 Module Perspective

1.5 Document Overview

Sections 1 and 2 provide an overview of the Rotation Utility Module, summarizing the purpose, services, context, and interfaces of the module. Section 3 describes all software structures and constants visible to modules outside of the Rotation Utility Module. Section 4 describes the functions provided to other modules by the Rotation Utility Module. Section 5 describes some of the implementation details of the module. Section 6 describes which requirements are met by the module. Section 7 describes what types of tests need to be performed to ensure that the implementation of the module is operating properly.

1.6 Acronyms

Please refer to the **Flight Software Global Design Specification**, Section 1.5, for a list of acronyms and abbreviations.

2 Overview

2.1 Purpose

The purpose of the Rotation Utility Module is to provide functions that the Flight Software uses to manage ASM rotations.

2.2 Services

The Rotation Utility Module provides the following services:

1) Calculate Rotation Times .... The Rotation Utility Module provides the mechanism that the Flight Software uses to compute the time it takes for the ASM to perform a given rotation and settle.
2) Issue Rotation Commands .... The Rotation Utility Module provides the mechanism that the Flight Software uses to issue rotation commands to the ASM.

2.3 Module Context

The following illustrates the Rotation Utility Module relationships with the rest of the Flight Software residing in an EA:

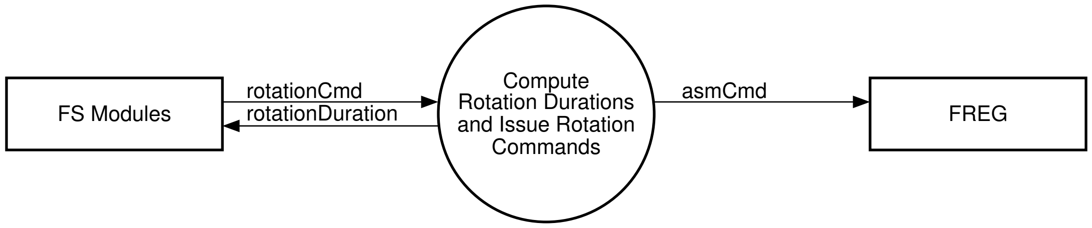

Figure 2 Rotation Utility Module Context Diagram

The Flight Software Modules use the Rotation Utility Module to compute the time it takes for the ASM to perform a given rotation (rotationDuration). They also use the Rotation Utility Module to form and issue the rotation commands (rotationCmd). The Rotation Utility Module uses the **Front Registers Module** to write ASM commands to the ASM Command Register (asmCmd).

2.4 Theory of Operation

The Rotation Utility Module provides two services to the rest of the Flight Software. To compute the duration of a given rotation, the caller passes the number of 1/2 degree steps to rotate. The Rotation Utility Module multiplies the passed value by a constant time/step value, adds a constant settling time, scales and rounds the result up to the nearest integer second, and returns the result to the caller. To rotate the cameras, the caller passes the number of 1/2 degree steps to rotate, and the direction of the rotation. The Rotation Utility Module then forms an ASM rotation command and writes the formed command to the ASM Command Register, using the **Front Registers Module**.

2.5 Interface Summary

The Rotation Utility Module uses **Front Registers Module** to write commands to the ASM Command Register

2.6 Data Structures and Constants Summary

The Rotation Utility Module uses the ASM Command register definitions provided by **ASM/EDS Interface Control Document**.

2.7 Function List Summary

This module defines the following functions:

ROTUTILduration()	Compute the time taken to perform a rotation.
ROTUTILrotate()	Issue an ASM Rotation command.

3 Data Structures and Constants

3.1 Hardware

3.1.1 ASM Rotation Commands

The ASM interprets the 16-bit ASM Command Word as follows (see **Front Registers Module** and **ASM to EDS Interface Control Document**):

Command Tag (bit 0) .... Set to zero to indicate a command.
Component Id (bits 1-2) .... Must be zero to address the drive assembly.
Step Size (bit 3) .... Rotation Utility Module always sets this bit to 1 to perform 1/2 degree rotation steps.
Steps (bits 4-14) .... This is the number of 1/2 degree steps to perform.
Direction (bit 15) .... This bit specifies the direction of the rotation.
0 .... Clockwise (away from Home).
1 .... Counter-Clockwise (toward Home).

3.2 Constants

3.2.1 ROTATE_MAX_STEPS (2047)

This is the maximum number of 1/2 degree steps supported by the ASM-EDS command interface (see **ASM to EDS Interface Control Document**, "Command Format: Rotation").

3.2.2 ROTATE_TIME_SCALE (128)

This constant defines the time scale, in fractions of a second, used by this module for its internal time constants and computations (see Sections 3.2.3, 3.2.4 and 5.2).

3.2.3 ROTATE_STEP_TIME (43)

This is the time it takes for the ASM to perform one 1/2 degree step, in increments of 1/ROTATE_TIME_SCALE of a second (see Section 3.2.2).

3.2.4 ROTATE_SETTLE_TIME (192)

This is the time it takes for the ASM to settle after a rotation has finished, in increments of 1/ROTATE_TIME_SCALE of a second (see Section 3.2.2).

3.2.5 ROTATE_STEP_TIME_PATCHED_FOR_SIDE_A (61)

This value is provided as a reference for post-launch maintenance of the Flight Software, and is not used directly within the delivered software. It represends the time it takes for the ASM Side-A windings to rotate the ASM by one 1/2 degree step, in increments of 1/ROTATE_TIME_SCALE of a second (see Section 3.2.2).

3.2.6 MSG_ROTUTIL_DURATION_STEPS

This message code is passed to UTILpanic() if ROTUTILduration() is passed a step argument greater than ROTATE_MAX_STEPS. This message should read something like:

"PANIC: --ROTUTILduration(%d) --Steps argument too large"
where %d represents the passed "*steps*" argument.

3.2.7 MSG_ROTUTIL_ROTATE_STEPS

This message code is passed to UTILpanic() if ROTUTILrotate() is passed a step argument greater than ROTATE_MAX_STEPS. This message should read something like:

"PANIC: --ROTUTILrotate(%d) --Steps argument too large"
where %d represents the passed "*steps*" argument.

4 Functions

4.1 ROTUTILduration() - Compute Rotation Time

Usage

UINT16 ROTUTILduration (UINT16 *steps*);

Description

ROTUTILduration() computes the number of seconds it takes to rotate the ASM the number of 1/2 degree output rotation steps specified by "*steps*", using a constant rotation rate, ROTATE_STEP_TIME. This function adds a constant value, ROTATE_SETTLE_TIME, to the result to accommodate the settling time of the rotation assembly.

If "*steps*" is greater than ROTATE_MAX_STEPS, ROTUTILduration() will issue a MSG_ROTUTIL_DURATION_STEPS panic message and reset the EA, passing "*steps*" as the message argument, using the **Utility Module** function UTILpanic().

Return Values

Number of seconds that it takes to perform a given rotation and settle.

Caveats

1) ROTUTILduration() always rounds up to the next integer when computing the rotation time.

See Also

ROTUTILrotate(), **Utility Module Specification**

4.2 ROTUTILrotate() - Issue an ASM Rotation Command

Usage

void ROTUTILrotate (UINT16 *steps*, BOOL *direction*);

Description

ROTUTILrotate() forms an ASM rotation command and sends the command to the ASM. "*steps*" is the number of 1/2 degree output rotation steps of the drive to rotate. If "*direction*" is TRUE, the cameras will rotate in a clockwise direction (away from the Home Position). If "*direction*" is FALSE, the cameras will rotate in a counter-clockwise direction (toward the Home Position). If "*steps*" is zero, no rotation command will be issued.

If "*steps*" is greater than ROTATE_MAX_STEPS, ROTUTILrotate() will issue a MSG_ROTUTIL_ROTATE_STEPS panic message and reset the EA, passing "*steps*" as the message argument, using the **Utility Module** function UTILpanic().

Return Values

NONE - FUNCTION IS VOID

Caveats

1) The caller must call ROTUTILduration() to obtain the number of seconds it takes to perform the operation, and must wait this amount of time before issuing another rotation command to the ASM.

2) The caller must ensure that at least 4ms elapse after calling this function before issuing a housekeeping query to the ASM. It must also ensure that at least 6ms elapse after issuing a query before calling this function (see **Housekeeper Module Specification**, "Control Considerations : Coupling with Position Histogram Science Mode"). Please refer to the **ASM to EDS Interface Control Document**, "Command Bus Protocol" and "Housekeeping Query Protocol" for a description of the temporal requirements of the ASM to EDS Interface.

3) This function will destroy any unread query response from the ASM Housekeeping register.

See Also

ROTUTILduration(), **Utility Module Specification**

5 Detailed Design Considerations

5.1 Data Flow

The following illustrates the Level 1 data flow within the Rotation Utility Module:

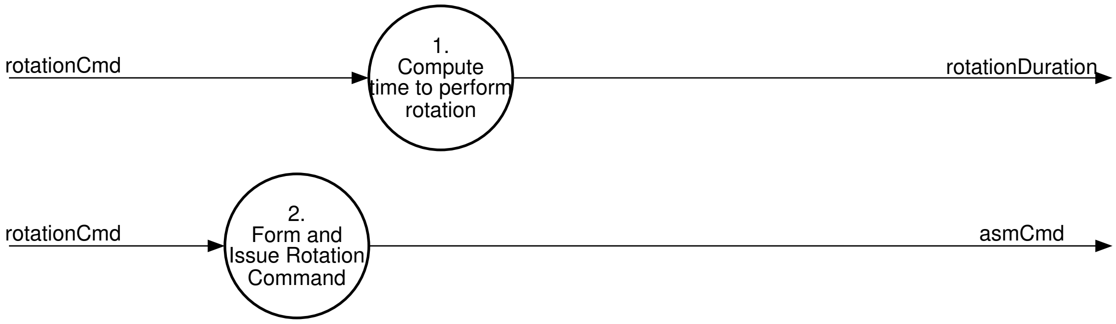

Figure 3 Rotation Utility Module Level 1 Data Flow

Flight Software Modules use the Rotation Utility Module to calculate the amount of time a given ASM rotation will take. The caller passes the number of steps to perform for the rotation (rotationCmd) and the Rotation Utility Module computes (bubble 1) the time it will take to perform the specified rotation and returns the result to the caller (rotationDuration).

The Flight Software Modules use the Rotation Utility Module to issue rotation commands to the ASM. The caller passes the number of 1/2 degree steps to perform, and the direction of the rotation. The Rotation Utility Module forms the ASM Command word (bubble 2) and sends the command to the ASM (asmCmd).

5.2 Rotation Time Computation

The Rotation Utility Module computes the time it takes to perform a given rotation according to the following formula:

$$time = ((steps * \mathrm{ROTATE\_STEP\_TIME}) + \mathrm{ROTATE\_SETTLE\_TIME}) / \mathrm{ROTATE\_TIME\_SCALE}$$

where "*steps*" is the number of 1/2 degree steps to rotate, and "*time*" is the time it will take to perform the rotation (in seconds). The result of the division by

ROTATE_TIME_SCALE is always rounded up to the nearest integer second (NOTE: This may be accomplished by always adding 1 to the result of the division).

Since the rotation step time is specified in 1/ROTATE_TIME_SCALE second increments, and the ROTATE_STEP_TIME is always rounded up from the actual value, the maximum possible error per step is 1/ROTATE_TIME_SCALE seconds. Given the maximum number of steps that can be specified, this leads to a maximum possible over-estimated rotation time of ROTATE_MAX_STEPS/ROTATE_TIME_SCALE seconds, plus or minus a second (i.e. about 16 seconds).

5.3 Module Structure

The Rotation Utility Module does not use any internal functions. The Rotation Utility Module does, however, call the **Front Registers Module** function, FREGwrite_asm, to write the ASM Command to the ASM's Command Register. For more detail on the operation of the Rotation Utility Module, refer to the module's detailed design specification.

6 Requirements Traceability

6.1 Science Requirements

The Rotation Utility Module provides mechanisms used by the Flight Software to issue rotation commands to the ASM, and to compute the rotation time so that it can ensure that the cameras are motionless while performing data accumulation.

Please refer to the **Requirements Traceability Matrix** for a list of the specific requirements met by this module.

7 Testing Considerations

7.1 Unit Testing

To perform unit tests on this module, the following equipment is required:

a. An Event Analyzer equipped with the Intermetrics ROM Monitor

b. A workstation running the Intermetrics debugger

c. An RS-232 serial cable between the ROM Monitor's serial port and the workstation

d. A logic analyzer

To unit test this module, the tester must link and locate this module's functions to some EA RAM memory location. Using a resident monitor/debugger, the tester should first verify the operation of all functions provided by this module, and then verify the interactions between all functions provided by the module. These tests should verify all branches, calls, and legal and illegal parameters.

In addition to these tests, the tester should verify the format of the ASM Command Word for camera rotations, and verify the computed rotation/settling time against the mechanical specifications.

7.2 Integration Testing

For integration tests, the following equipment is required:

a. An Event Analyzer

b. A populated ROM Repository board

c. A logic analyzer

d. Hardware which can:

   1. Rotate in response to ASM commands (ASM Motor Assembly/Electronics).

The integration tests should verify the operation of the Rotation Utility Module in conjunction with the actual hardware and other FS modules. These tests should include:

a. Verify that the rotation commands are correctly formed and sent to the ASM and that the ASM performs the specified rotations.

b. Verify that the computed rotation and settling time is greater than or equal to the actual rotation and settling time in both rotation directions and using a variety of rotation angles.

Appendix A

A.1 Items Under Review

This section lists items and issues which need to be resolved:

Document Reference	Section	Resolution Expected by
NONE		