Housekeeper Module Specification

James Francis

MIT 64-30112.0201

Rev. C

August 11, 1993

Contents

Figures

Tables

1 Introduction

1.1 Scope

This document describes the Housekeeper Module (HOUSE). The Housekeeper Module consists of a thread and a MinorTick Interrupt client function which are responsible for controlling the acquisition of housekeeping data from the All-Sky Monitor's (ASM) housekeeping sensors, and for posting the acquired data for downlink.

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 64-02006.01 Rev. D .... "XTE EDS Hardware Functional Description and Requirements"
MIT 64-03006 Rev. B .... "ASM to EDS Interface Control Document"
MIT 64-30102 Rev. E....................... "Flight Software Global Design".
MIT 64-30110.0101 Rev. B .... "Back Registers Module Specification" (BREG)
MIT 64-30110.1201 Rev. D .... "Interrupt Manager Module Specification" (INTR)
MIT 64-30110.0401 Rev. C .... "Downlink Module Specification" (DOWNLINK)
MIT 64-30110.0701 Rev. C .... "Timing Module Specification" (TIMING)
MIT 64-30110.0901 Rev. C .... "Front Registers Module Specification" (FREG)
MIT 64-30110.1701 Rev. D .... "Thread Manager Module Specification" (THREAD)
MIT 64-30110.1901 Rev. D .... "Uplink Module Specification" (UPLINK)
MIT 64-30110.2001 Rev. C .... "Utility Module Specification" (UTIL)
MIT 64-30111.0201 Rev. C .... "Buffer Pool Module Specification" (BUFP)
MIT 64-30112.0601 Rev. D .... "Science Thread Module Specification" (SCIENCE)
MIT 64-30112.060108 Rev. C .... "Position Histogram Mode Module Specification" (POSITION)

1.3 Notations

This document assumes the following notations:

UINT16 - Refers to a 16-bit unsigned value.

UINT32 - Refers to a 32-bit unsigned value.

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 Housekeeper Module's scope is indicated by the shaded area:

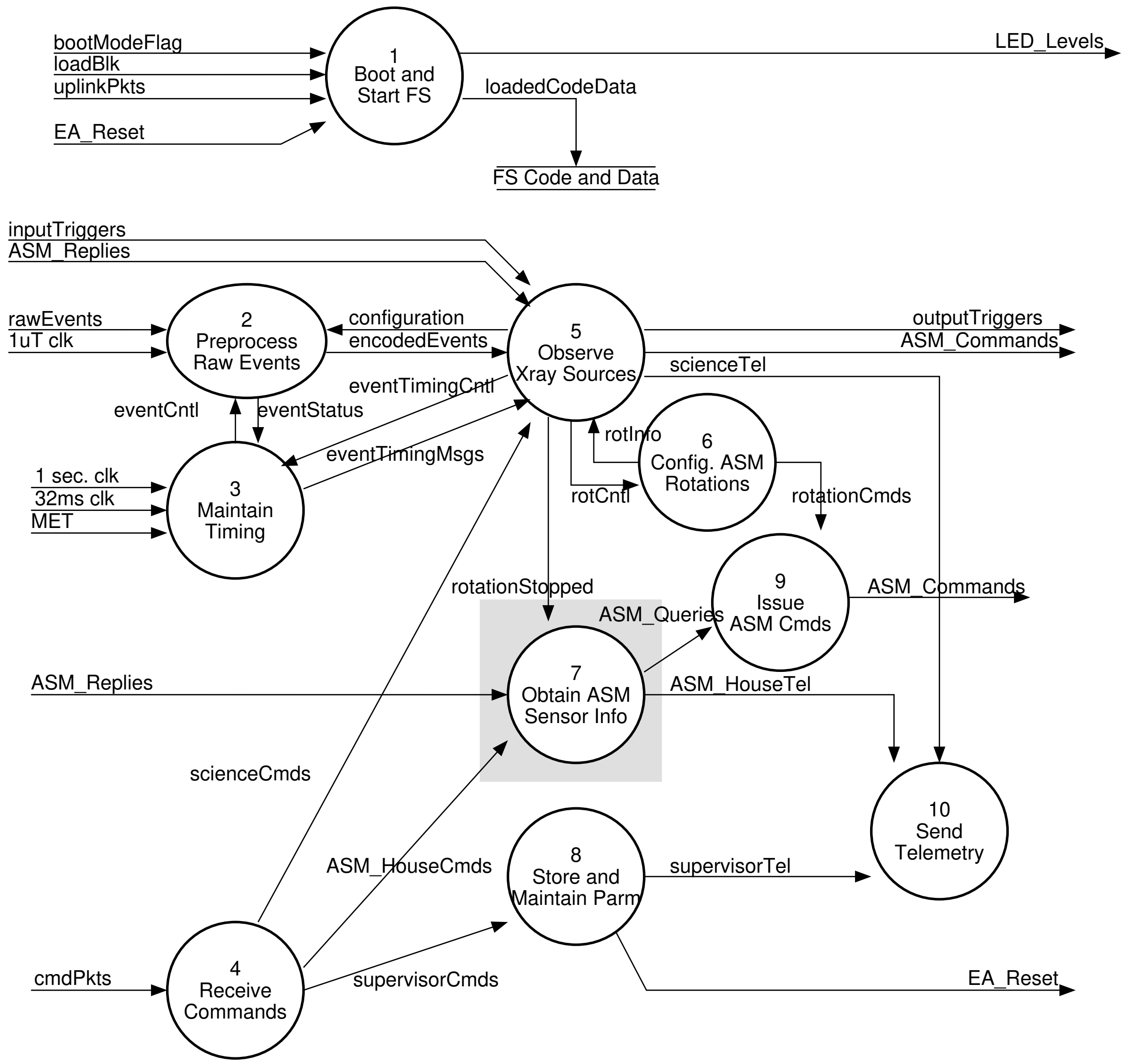

Figure 1 Module Perspective

1.5 Document Overview

Sections 1 and 2 provide an overview of the Housekeeper Module, summarizing the purpose, services, context, and interfaces of the module. Section 3 describes all software data structures and constants visible to modules outside of the Housekeeper Module. Sections 4 and 5 describe the interrupt and the event notifications used by the Housekeeper Module. Section 6 describes the module initialization function provided by the Housekeeper Module. Section 7 describes some of the implementation details of the module. Section 8 describes which requirements are met by the module. Section 9 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 Housekeeper Module is to obtain and telemeter engineering housekeeping data from the ASM Engineering Data Sensors.

2.2 Services

The Housekeeper Module provides the following services:

1) Collect ASM Sensor Values .... The Housekeeper Module is responsible for collecting the ASM Sensor values. It does this by issuing queries to the ASM's drive and camera subassemblies and retrieving and storing the responses to these queries.

2) Telemeter Collected Values .... The Housekeeper thread is responsible for copying the collected ASM Sensor values into telemetry packets, and for posting the formed packets for transfer out of the EDS.

2.3 Module Context

The following illustrates the Housekeeper Module's relationships with the rest of the Flight Software residing in an EA:

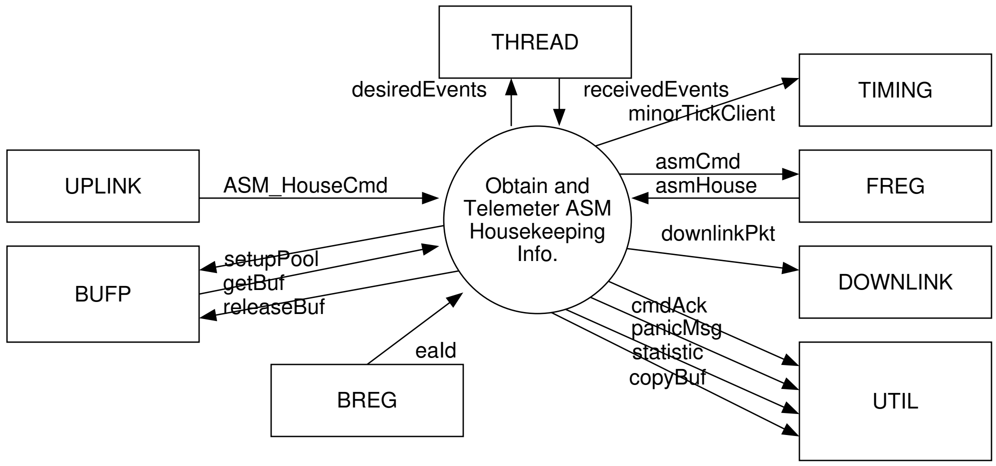

Figure 2 Housekeeper Module Context Diagram

When the Housekeeper Thread's installation routine is invoked during startup, it uses the **Back Registers Module** to obtain the Event Analyzer (EA) identifier. If the EA is the Sub-Group B All-Sky-Monitor EA (ASM-EA, Side-B), the installation routine proceeds to install the Housekeeper Thread, which is run once multitasking starts. If the EA is not an ASM-EA, the Housekeeper Thread is not installed, and will not run.

Once the Housekeeper Thread starts, it installs a MinorTick Interrupt client function (minorTickClient) using the **Timing Module**, and sets up a pool of telemetry buffers (setupPool), using the **Buffer Pool Module**. The **Timing Module** calls the installed client function while handling MinorTick interrupts (~32ms).

While running, the Housekeeper Module uses the **Thread Manager Module** to wait for one or more event notifications (desiredEvents) and to obtain the set of events which were notified (receivedEvents). When commands arrive, the Housekeeper Module uses the **Uplink Module** to obtain command packets addressed to the Housekeeper Thread (ASM_HouseCmd), the **Utility Module** to generate acknowledgments to the commands (cmdAck), and the **Buffer Pool Module** to release the command's packet buffer back to its pool (releaseBuf). Although the Housekeeper Module does not execute any commands, it must provide facilities to remove command packets addressed to it from the uplink queue, to allow subsequent commands to be retrieved by the other threads (see **Flight Software Global Design**, "Design Description, Threads").

The Housekeeper Module's MinorTick client function uses the **Front Registers Module** to issue ASM query commands (asmCmd) and to obtain the ASM's response to these commands (asmHouse). Once a set of housekeeping information has been acquired, the Housekeeper Module obtains a telemetry buffer (getBuf), using the **Buffer Pool Module**, copies the obtained housekeeping information into the buffer, using the **Utility Module** (copyBuf), and posts the formed telemetry packet (downlinkPkt) for transfer out of the EA using the **Downlink Module**.

The Housekeeper Module also uses the **Utility Module** to issue panic messages (panicMsg), and report software housekeeping statistics (statistic).

2.4 Theory of Operation

The Housekeeper Module is responsible for acquiring and telemetering a complete set of ASM Housekeeping information for each partition of data produced by the **Position Histogram Science Mode** (see **Science Thread Module Specification**, and **Position Histogram Mode Module Specification**). The Housekeeper Module consists of a Housekeeper Thread and a MinorTick Interrupt Client function. The Housekeeper Thread is only run on the primary ASM-EA (ASM-EA Side-B), since this is the only EA with access to the ASM Housekeeping interface (see **XTE EDS Hardware Functional Description and Requirements**, "FREGS and Observation Control - Command/Housekeeping Register").

During system initialization, before multitasking starts, the **Startup Module** invokes the initialization function for the Housekeeper Module, which then uses the **Back Regisers Module** to obtain the EA Id. If the EA Id indicates the Subgroup B ASM-EA, the Housekeeper Module registers the thread's context with the **Thread Manager Module**. Once multitasking starts, the thread is invoked by the **Thread Manager**. When the Housekeeper thread is run, it initializes its state variables, installs the Housekeeper Module's MinorTick Interrupt Client function, and waits for notification from the **Position Histogram Science Mode** that the ASM Command/Housekeeping Interface is available for use (i.e. a rotation is not in progress). If **Position Histogram Mode** isn't run on the EA, the Housekeeper Thread will remain suspended indefinitely.

As **Position Histogram Mode** starts each dwell it notifies the Housekeeper Thread that the ASM Housekeeping interface is available for use by the Housekeeper Thread. When the Housekeeper Thread receives the event notification, the Housekeeper Thread sets a flag which enables the MinorTick Interrupt Client function's data acquisition process.

Upon the MinorTick Interrupt after the flag is set, the client function issues its first query to the ASM and resets the flag. Upon the next MinorTick Interrupt, the client function reads the response from the first query, storing the read value in an array of housekeeping values, and issues the next query. This process of reading responses and issuing queries repeats for each MinorTick Interrupt until the entire set of housekeeping values have been read. Once the last value has been read and stored, the client function stops issuing queries and notifies the Housekeeper Thread that a complete set of housekeeping values has been accumulated.

When the Housekeeper Thread receives the notification that a complete set of data has been accumulated, it obtains a telemetry buffer, copies the accumulated data into the buffer, and posts the buffer to be transferred out of the EA. The Housekeeper Thread then waits for another notification from the **Position Histogram Mode**. When the next event notification is received, it again sets the flag which starts data accumulation and the process repeats. Once **Position Histogram Mode** ends or is terminated, it stops sending event notifications to the Housekeeper Thread, which ends up suspended until the EA is reset.

2.5 Interface Summary

The Housekeeper Module directly uses the services provided by the following modules:

THREAD Module .... Notification of desired Events.
UPLINK Module............................ Provides Command Packets.
BUFP Module................................. Provides ASM Housekeeping telemetry buffers
Releases Command Packet Buffers to appropriate pool.

TIMING Module .... Installs MinorTick Interrupt Client Function.
Calls installed client function on every MinorTick Interrupt
BREG Module .... Provides the Event Analyzer Identifier
FREG Module .... Writes ASM Command Register.
Reads ASM Housekeeping Register.
DOWNLINK Module .... Posts telemetry packets for transfer out of an EA.
UTIL Module .... Posts panic messages and resets the EA.
Issues Command Acknowledgments.
Performs fast memory copies.
Records Software Housekeeping Statistics.

The Housekeeper Module indirectly relies on actions performed by the following:

POSITION Module .... Notifies the Housekeeper Thread with an event at the start of each dwell, indicating that the ASM Command/Housekeeping interface is available for use.

2.6 Data Structures and Constants Summary

The Housekeeper Thread uses the hardware bit definitions of the ASM Command register, and the association between the query commands and the results read from the ASM Housekeeping register. It also defines the label and structure of the housekeeping information that is sent to the ground. This thread also defines the command codes and arguments that it responds to, and the structure of the commands.

2.7 Event Summary

The Housekeeper Thread may wait for the following events:

EV_NEW_UPLINK .... A new command packet has been received for this thread.
EV_ASM_IF_AVAIL .... Indicates that the Housekeeper can start issuing query requests to the ASM engineering sensors.
EV_ASMHOUSE_RDY .... The Housekeeper Module's MinorTick Interrupt client function notifies the Housekeeper Thread with this event when a complete ASM Housekeeping Data Set is ready to be telemetered.

2.8 Interrupt Summary

The Housekeeper Thread installs its own MinorTick Interrupt client function, using the **Timing Module**. This function must be called once every MinorTick interrupt (~32ms).

2.9 System Resource Utilization

telemetry .... This module produces command acknowledgments for each command received. This has a very minor impact on telemetry utilization. It also generates ASM Housekeeping Telemetry packets at a regular rate, governed by the dwell/rotation rate of the **Position Histogram Science Mode**. Assuming that the worst case is one packet every 10 seconds, where each packet is 132 bytes + 16 bytes of packet overhead (total of 1184 bits), this module will use a maximum of 4% of the telemetry allocated to the ASM-EA's.

interrupt latency .... The MinorTick Interrupt client function, installed by this module, doesn't do a huge amount of work. It should have a minor impact on interrupt latencies of the MinorTick Interrupt and interrupts with a lower priority (than the MinorTick Interrupt).

3 Data Structures and Constants

3.1 Hardware

3.1.1 ASM Command Register (FREG)

The Housekeeper Module writes query commands to the ASM Command Register using the **Front Registers Module**. The format and protocol requirements for query commands are defined in the **ASM to EDS Interface Control Document**, "Command Format", "Housekeeping Query Protocol" and "Housekeeping Query Format".

The following bit definitions are used for the detector queries:

HOUSE_QUERY_TAG .... Bit 0 .... Always set to one for ASM Query Commands.

HOUSE_QUERY_DEV_ID .... Bits 1 and 2 .... Specifies which device is being queried. The values are as follows:
00 ...... Drive Assembly
01 ...... Scanning Shadow Camera 1
10 ...... Scanning Shadow Camera 2
11 ...... Scanning Shadow Camera 3

HOUSE_QUERY_INFO_ID ...... Bits 3-6 .... Specifies which information channel to query on the addressed device (see HOUSE_QUERY_DEV_ID). The meaning of each Information Channel is dependent on the device being addressed. Refer to the **ASM to EDS Interface Control Document**, "Housekeeping Data Format", for the description of each Information Channel.

HOUSE_QUERY_UNUSED ..... Bits 7-15 .... These bits are unused, and are set to zero by the Housekeeper Module.

3.1.2 ASM Housekeeping Register (FREG)

The Housekeeper Module reads query responses from the ASM Housekeeping Register using the **Front Registers Module**. The format of the responses are defined in the **ASM to EDS Interface Control Document**, "Housekeeping Data Format." The Housekeeper Module treats each response as a raw 16-bit word associated with the query command which requested the value. It does not perform any further interpretations of these values. It is the ground’s responsibility to interpret the meaning of each response.

3.2 Data Structures

3.2.1 Housekeeping Telemetry Packet

As for all telemetry packets produced by the EA's, the ASM Housekeeping Telemetry packet consists of a CCSDS header, followed by a secondary header, followed by the packet's stream id, followed by a single reserved byte, followed by stream specific data. For a description of the CCSDS and secondary headers, and of the location of the stream id field, refer to the **Flight Software Global Design Specification**, Section 3.2.2. The stream id of ASM Housekeeping Telemetry packets is described in Section 3.3.2. The stream specific data is as follows:

[sequence #...] .... ASM Housekeeping Sequence # (4 bytes)
This value starts at zero and is incremented for each set of housekeeping data acquired from the ASM. There should be a one-to-one correspondence between each set and each partition of data sent by the Position Histogram Science Mode. Discarded or missing sets are indicated by gaps in the sequence numbers. The Housekeeper Module indicates discarded data sets in the Software Housekeeping Data Base.

[housekeeping data...] .... 64 16-bit ASM Housekeeping Values (128 bytes)
This is an array of 64 16-bit values obtained from the ASM. The array's values are indexed by ASM Device Id and Information Channel.
For example:
Word 0 .... corresponds to device 0, channel 0
Word 1 .... corresponds to device 1, channel 0
Word 2 .... corresponds to device 2, channel 0
Word 3 .... corresponds to device 3, channel 0
Word 4 .... corresponds to device 0, channel 1
...
Word 7 .... corresponds to device 3, channel 1
Word 8 .... corresponds to device 0, channel 2
...
Word 62 .... corresponds to device 2, channel 15
Word 63 .... corresponds to device 3, channel 15

3.3 Constants

3.3.1 THREAD_ID_ASMHOUSE

This value uniquely identifies the Housekeeper Thread on a given EA. This value is used for the following:

a. Used by the Thread Manager Module to identify the thread (see **Thread Manager Module Specification**)

b. As the "*Destination Thread Id*" in Housekeeper Command Packets. Currently the Housekeeper Thread does not execute any commands. If the ground sends a command to the Housekeeper Thread to test that the thread is running, the Housekeeper Thread will generate a CMD_REPLY_UNKNOWN reply to the command (see **Utility Module Specification**, "Functions : UTILack()").

c. Provided in all Command Acknowledgment packets produced by the thread.

d. Used by the **Position Histogram Science Mode** when notifying the Housekeeper that the ASM's Command/Housekeeping Interface is available for use (see Section 5).

3.3.2 STREAM_ID_ASM_SENSOR

This value is stored in the "*Data Stream Id*" of a telemetry packet to indicate that the packet contains ASM Housekeeping Data (see Section 3.2.1).

3.3.3 HOUSE_NUM_DEVICE_IDS (4)

This constant defines the number of ASM Device Id's.

3.3.4 HOUSE_MAX_CHANNELS (16)

This constant defines the maximum number of ASM Information Channels for any given ASM Device Id.

3.3.5 HOUSE_MAX_GATHER_TIME (3 seconds)

This constant defines the maximum amount of time, rounded up to the nearest second, that it takes for the Housekeeper Module to acquire all of the housekeeping data from the ASM. It is defined using the following formula:

((ASMHOUSE_NUM_DEVICE_IDS * ASMHOUSE_MAX_CHANNELS) + 1) * 32ms

NOTE: This value does not take into account the delay between the time **Position Histogram Mode** notifies the Housekeeper Thread that the ASM Interface is available and the instant at which the Housekeeper Thread's MinorTick client function issues the first query to the ASM.

3.3.6 HOUSE_QUERY_TAG (1)

This constant is "or'd" with an ASM Command Word to indicate that the word is a Query Command.

3.3.7 HOUSE_QUERY_DEV_ID_SHFT (1)

This constant defines the bit position of the device id within an ASM Query Command Word.

3.3.8 HOUSE_QUERY_MASK (0x0007f)

This value masks off all unused bits within an ASM Query Command Word.

3.3.9 HOUSE_DEFAULT_DATA (0x055)

Upon startup, the Housekeeping Thread fills its housekeeping buffer with this byte value. This will cause unfilled housekeeping device/information entries to contain the 16-bit word value 0x05555.

3.3.10 HOUSE_QUERY_COUNT (64)

This value defines the number of acquired housekeeping device/information entries. It is derived from HOUSE_NUM_DEVICE_IDS (4) * HOUSE_MAX_CHANNELS (16). See Sections 3.3.3 and 3.3.4.

3.3.11 HOUSE_DATA_LEN (128)

This value defines the number of bytes of housekeeping data sent with each ASM Housekeeping Telemetry Packet. This value does not include the length of the sequence number.

3.3.12 HOUSE_STACK_SIZE (1024)

This value defines the number of words of stack space reserved by the Housekeeper Thread.

3.3.13 HOUSE_BUFFER_POOL_SIZE (1036)

This value defines the number of bytes in the Housekeeper Thread's telemetry buffer pool.

3.3.14 CMD_REPLY_UNKNOWN

Since the Housekeeper Module does not currently interpret or execute any commands, it sends this reply code with all command acknowledgments.

3.3.15 MSG_HOUSE_ENABLE_CLIENT_ALREADY

This message code is passed to UTILpanic(), if the internal function house_enable_client() detects that the MinorTick client function is already accumulating housekeeping data. This error can occur if the module's state variables have been corrupted, or if the dwell interval is too short. The message should read something like:

"PANIC: ---house_enable_client() --Client function already accumulating data"

3.3.16 MSG_HOUSE_SEND_NO_DATA

This message code is passed to UTILpanic(), if the internal function house_send() detects that the MinorTick client function wasn't accumulating housekeeping data. This error can occur if the module's internal state variables have been corrupted. The message should read something like:

"PANIC: ---house_send() --Client function not accumulating data"

3.3.17 MSG_HOUSE_MINOR_CLIENT_ALREADY

This message code is passed to UTILpanic(), if the MinorTick Client function house_minor_client() is told to start accumulating a data set while it is already accumulating another data set. This error can occur if the module's internal state variables have been corrupted. The message should read something like:

"PANIC: ---house_minor_client() --Already accumulating data"

3.3.18 MSG_HOUSE_MINOR_CLIENT_QUERY

This message code is passed to UTILpanic(), if the MinorTick Client function house_minor_client() detects an invalid query identifier. This error can occur if the module's internal state variables have been corrupted. The message should read something like:

"PANIC: ---house_minor_client(%d) --Bad Query Id"
where %d is the offending query identifier

4 Interrupts

4.1 MinorTick Interrupt (Level 4)

The Housekeeper Module installs a MinorTick Interrupt Client Function, using the **Timing Module**. This client function is invoked by the **Timing Module** while processing MinorTick Interrupts, which are generated by the MinorTick clock about once every 32ms.

5 Events

5.1 EV_NEW_UPLINK

The **Uplink Module** notifies the Housekeeper thread with this event when that module receives and adds a new uplinked command packet to the thread's command queue.

5.2 EV_ASM_IF_AVAIL

This event is posted by the **Position Histogram Mode Module**. This event is posted to the Housekeeper thread at the beginning of every dwell phase of a dwell sequence and indicates that the ASM's Command/Housekeeping Interface is available for use.

5.3 EV_ASMHOUSE_RDY

This event is posted by the Housekeeper Module's MinorTick Interrupt Client Function once a complete set of ASM Housekeeping data has been accumulated.

6 Functions

6.1 HOUSEinit() - Initializes ASM Housekeeper Thread Context

Usage

void HOUSEinit (void);

Description

HOUSEinit() initializes the context of the ASM Housekeeper Thread, and is intended to be called once by the STARTUP module during system initialization. HOUSEinit() uses the **Back Registers Module** to get the Event Analyzer Id. If the Event Analyzer Id indicates the Subgroup B ASM-EA, HOUSEinit() uses the **Thread Manager Module** to construct the thread's descriptor, and add the descriptor to the system's list of active threads. If the EA is a PCA-EA, or is the ASM-EA in Subgroup A, HOUSEinit() does not install the Housekeeper Thread, thus preventing the thread from being run.

Return Values

NONE - FUNCTION IS VOID

Caveats

1) This function must be called once, during system initialization.

2) This function must be called after the **Thread Manager Module** has been initialized.

See Also

Thread Manager Module Specification, Startup Module Specification, **Back Registers Module**.

7 Detailed Design Considerations

7.1 Data Flow

The following illustrates the overall data flow within the Housekeeper Module (NOTE: The use of the EA Id during thread installation is not shown):

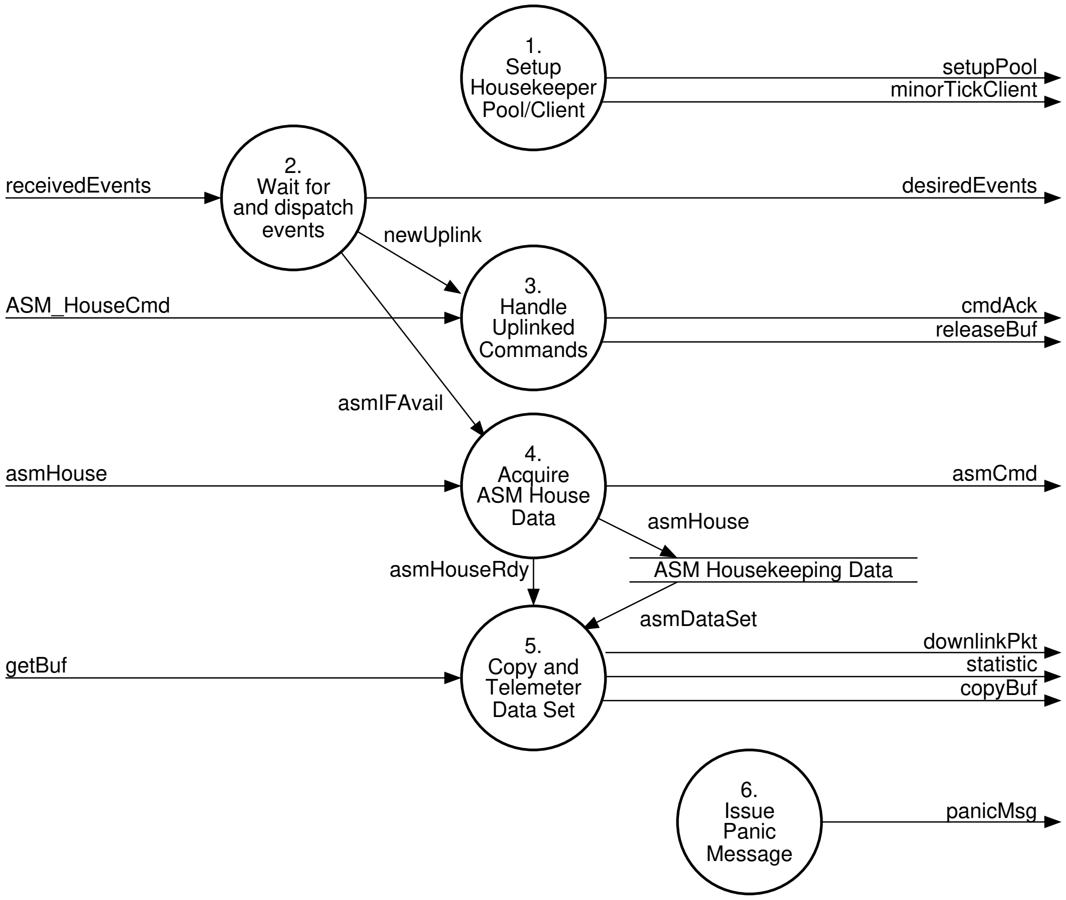

Figure 3 Housekeeper Module Data Flow

When the running Housekeeper Thread first gets control (bubble 1), it initializes its state variables (not shown), sets up a telemetry buffer pool (setupPool), and installs its MinorTick Client function (minorTickClient). It then proceeds to wait (bubble 2) for one or more event notifications (desiredEvents). When one or more of the desired event notifications have been received (receivedEvents), the Housekeeper handles each received event.

If an event is received which indicates that an uplink packet has arrived for the thread (newUplink), the Housekeeper (bubble 3) reads the command packet (ASM_HouseCmd), acknowledges the command (cmdAck) and returns the command packet's buffer to its original buffer pool (releaseBuf).

Each time the **Position Histogram Science Mode** finishes a rotation, or starts its first dwell, it notifies the Housekeeper Thread that the ASM Command/Housekeeping Interface is available for use (see **Position Histogram Mode Module Specification**, "Theory of Operation"). When the Housekeeper Thread receives the event notification (asmIFAvail), it enables the accumulation of housekeeping information from the installed MinorTick Interrupt Client Function. When the next MinorTick Interrupt occurs and the client function is invoked (bubble 4), the client function issues the first query command to the ASM (asmCmd), and returns. On the next MinorTick Interrupt, it reads the response to the query (asmHouse), stores the result in an array (ASM Housekeeping Data), and increments an index into the array. It then issues the next query (asmCmd) and returns. It repeats this process of reading and storing the responses and issuing the subsequent queries until it has issued the last query. During the subsequent MinorTick Interrupt, it reads the last response, and notifies the Housekeeper Thread that the ASM Housekeeping data set is complete (asmHouseRdy). The MinorTick Interrupt Client function is then disabled until another rotation has completed and the Housekeeper Thread enables housekeeping accumulation.

When the Housekeeper Thread (bubble 5) is notified that the ASM Housekeeping data set is complete (asmHouseRdy), it attempts to obtain a telemetry buffer from its buffer pool (getBuf). If it cannot obtain a buffer, it records the occurrence in the Software Housekeeping data base (statistic), and discards the completed data set. If a buffer is obtained, the Housekeeper Thread copies the ASM Housekeeping data set (asmDataSet) into the telemetry buffer (copyBuf). It then posts the filled packet for transfer out of the EA (downlinkPkt).

If the Housekeeper Module detects an error (bubble 6), it issues a panic message and resets the EA (panicMsg).

7.2 Control Considerations

7.2.1 Coupling with Position Histogram Science Mode

To ensure that the ASM Command/Housekeeping Interface temporal requirements are met (see **ASM to EDS Interface Control Document**, "Command Bus Protocol", "Housekeeping Query Protocol"), the Housekeeper Module is coupled to the **Position Histogram Science Mode** dwell/rotation sequence, using the EV_ASM_IF_AVAIL event. The overall sequence of operations is as follows, where each time unit represents some variable length of time during which the specified operations take place:

Time	Position Histogram Mode	Housekeeper Thread	MinorTick Client
t1	Setup Mode	Wait	Idle
t2	Notify Housekeeper Thread		
t3	Start Dwell		
t4		Enable Acquisition	
t5			Issue First Query
t6			Read Response Issue Next Query
t7			Read Response Issue Next Query
t8			<repeat>
t9			Read Response Issue Last Query
t10			Read Last Response Notify Housekeeper Idle
t11		Copy and telemeter	
t12	Start Rotation		
t13	End Rotation Notify Housekeeper Thread		
t14	Start Dwell		
t15		Enable Acquisition	
t16			Issue First Query
t17			<same as above>
t18		Copy and telemeter	

Table 1 Housekeeper and Position Histogram Operation Sequence

After the Housekeeper Module's thread initializes its state variables and installs its MinorTick Client function, the client function is idle (i.e. it doesn't perform any operations when invoked by the MinorTick Interrupt handler) and the thread waits for an event notification that the ASM Command/Housekeeping Interface is available for use (t1). After the **Position Histogram Science Mode** has initialized itself (t1), and is ready to start its first dwell, it notifies the Housekeeper Thread (t2) that the ASM Interface is available. **Position Histogram Mode** then schedules its first dwell, and waits for the dwell to complete (t3).

When the Housekeeper Thread receives the notification that the ASM Interface is available, it sets a flag for the MinorTick Client Function, indicating that it should start acquiring housekeeping data (t4). When the next MinorTick Interrupt occurs, and the MinorTick client function is invoked, the client function issues the first query command to the ASM (t5) and returns. Each subsequent MinorTick Interrupt causes the the client function to read and store the response to the previous query, and issue the next query command (t6, t7, t8, t9). On the next MinorTick Interrupt after the client function has issued the query for the last housekeeping value (t10), the client function reads and stores the last value and notifies the Housekeeper Thread that the ASM Housekeeping Data Set is ready. It then stops issuing queries and reading responses (t10).

When the Housekeeper Thread receives the event notification that the ASM Housekeeping Data Set is ready, it obtains a telemetry buffer, copies the completed data set into the packet, and posts the packet for transfer out of the EA (t11).

Meanwhile, the **Position Histogram Science Mode** is accumulating science data during its dwell. When the dwell ends, it issues a rotation command to the ASM (t12). **Position Histogram Mode** is responsible for ensuring that the dwell is long enough to accommodate the time it takes for the Housekeeper Module to read all of the ASM Housekeeping values. (t12) minus (t3) must be greater than HOUSE_MAX_GATHER_TIME plus some overhead. Assuming that the Housekeeper Thread obtains control within one second after **Position Histogram Mode** starts the dwell and relinquishes control, and that it takes less than one second for the Housekeeper Thread to set the shared flag, and for the MinorTick client function to be invoked and issue the first query, the overhead time is less than 2 seconds. This leads to a minimum dwell time of (HOUSE_MAX_GATHER_TIME + 2) seconds.

Once the rotation ends (t13), the **Position Histogram Science Mode** notifies the Housekeeper Thread that the ASM Interface is available. It then proceeds to start the next dwell (t14). When the Housekeeper Thread receives the event notification (t15), it again enables housekeeping data acquisition by the MinorTick Client function, and the acquisition process repeats (t16-18).

7.3 Module Structure

The following diagram illustrates the overall structure of the Housekeeper Module. Bolded boxes with bolded text represent functions provided by modules other than the Housekeeper Module. The dotted line from HOUSEinit to house_thread illustrates that HOUSEinit installs house_thread as the first function to call when the Thread Manager schedules the thread for the first time. All other boxes represent functions which are private to the Housekeeper Module.

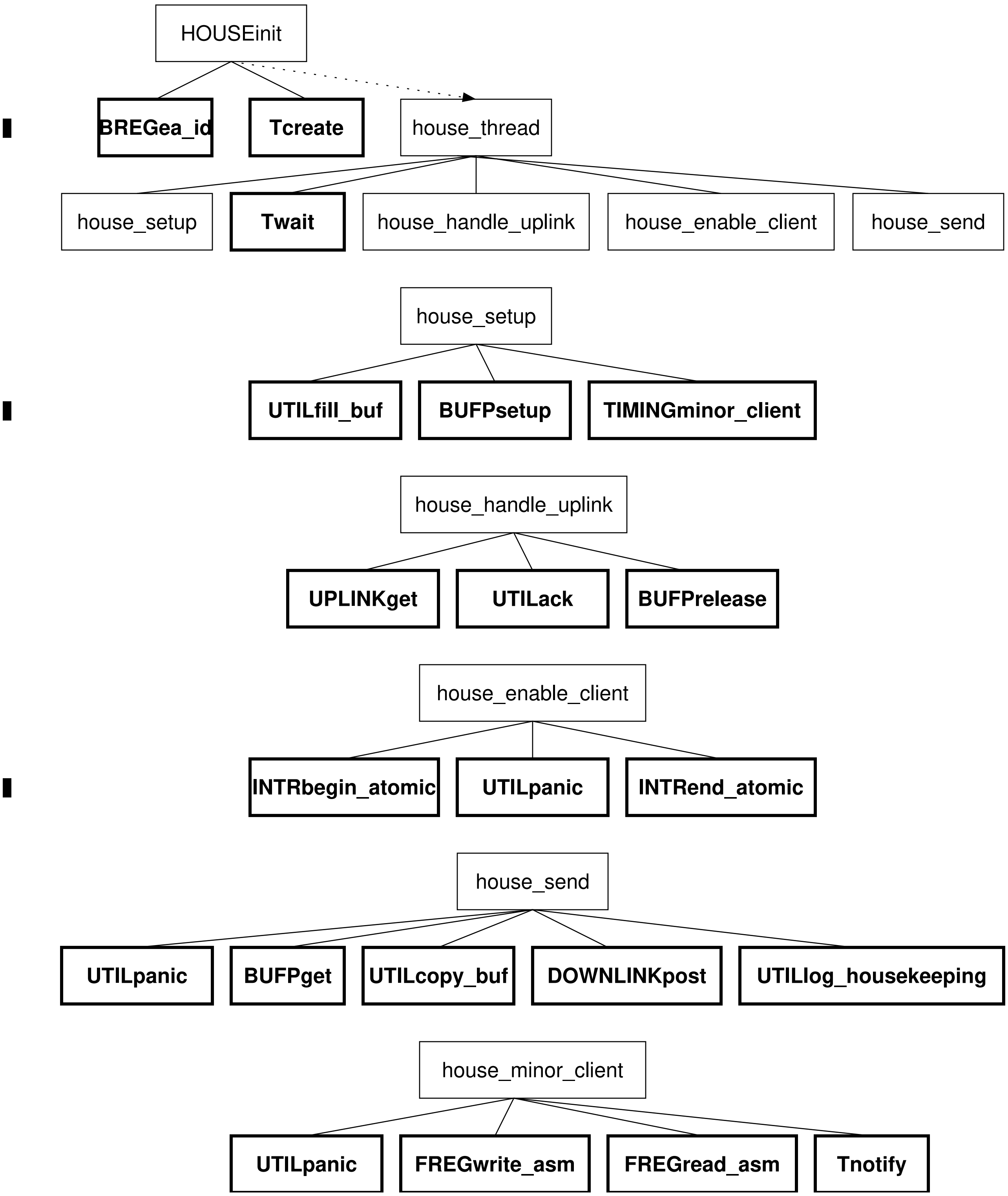

Figure 4 Housekeeper Module Function Structures

The following lists the functions shown in Figure 4. The details of each function's operation and arguments are described in the Housekeeper Module's detailed design document.

HOUSEinit	If on ASM-EA Side-B, install Housekeeper Thread.
house_thread	Dispatch control based on received event notifications.
house_setup	Initialize module state variables, buffer pool and install MinorTick client function.
house_handle_uplink	Handle any received command packets.
house_enable_client	Assert flag which enables the start of sensor data acquisition.
house_send	Form and post an ASM Housekeeping Telemetry Packet.
house_minor_client	Issue housekeeping queries and read and store responses.

8 Requirements Traceability

8.1 Science Requirements

The Housekeeper Module is responsible for satisfying the ASM Housekeeping Science Requirements, as specified in the **ASM Science Requirements Specification**. The specific requirements met by this module are indicated in the **Requirements Traceability Matrix**.

This module, in conjunction with the **Position Histogram Science Mode**, is also responsible for adhering to the ASM to EDS Interface functional and temporal requirements, as specified in the **ASM to EDS Interface Control Document**, "Command Bus Protocol", "Housekeeping Query Protocol", "Housekeeping Query Format", and "Housekeeping Data Format." Adherence to the format requirements can be verified by examining Sections 3.1.1 and 3.1.2. Assuming that the dwell times performed by the **Position Histogram Science Mode** are greater than the time it takes for the Housekeeper Module to read all of the ASM Housekeeping values, and that all rotation times are greater than 3.04ms (see **ASM to EDS Interface Control Document**, "Command Bus Protocol"), all temporal requirements are met by using the MinorTick Interrupt to govern the Housekeeper's access to the ASM's Command/Housekeeping registers. Since each write->write and write->read operation is separated by at least 32ms, all temporal requirements are met. There are no specified temporal requirements for read->write operations, although function call overhead and execution speed of the 80286 guarantees a minimum read->write time on the order of microseconds.

9 Testing Considerations

9.1 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 internal to this module, and then verify the interactions between all functions internal to the module. These tests should verify all branches, calls, and legal and illegal parameters. The tester should then verify all calls made to functions outside of this module, and ensure that the passed arguments are always within the limits specified by the corresponding module specifications. The tester should also verify that the caller always handles values returned by these functions in an appropriate manner.

In addition to these tests, the tester should verify that the Housekeeper Module properly forms the ASM Query Commands, properly issues the queries and obtains the responses for all device and information channels supported by the ASM, and stores the read responses in the proper location in the ASM Housekeeping Data Set. The tester should also verify that Housekeeping Module properly forms the ASM Housekeeping Telemetry packet.

9.2 Integration Testing

For integration tests, the following equipment is required:

a. An Event Analyzer

b. A populated System Manager board

c. A logic analyzer

d. Hardware which can:

   1. Send command packets to the EA (S/C Simulator)

   2. Receive telemetry packets via the RS422 Interface (S/C Simulator)

   3. Drive the MinorTick Interrupt Line (S/C Simulator)

   4. Block telemetry packets from an EA (i.e. stop receiving packets)

   5. Drive the ASM-EDS Command/Housekeeping Interface (ASM Simulator)

The integration tests should verify the operation of the Housekeeper Module in conjunction with the actual hardware and other FS modules, especially the **Position Histogram Science Mode**. These tests should include:

a. Verify that the Housekeeper Thread properly gets control after a reboot.

b. Verify that the Housekeeper Module obtains all ASM Housekeeping values within the minimum dwell time.

c. Verify that the Housekeeper Module properly forms and posts ASM Housekeeping Telemetry packets.

d. Verify that the Housekeeper Module and **Position Histogram Mode** don't contend for the ASM Command/Housekeeping registers.

e. Verify that the ASM Housekeeping Data set is accumulated while a dwell is being performed by the **Position Histogram Science Mode**.

f. Verify that all ASM Housekeeping Telemetry packets can be reliably associated with a corresponding Science Partition from the **Position Histogram Science Mode**.

g. Run **Position Histogram Mode** to completion and ensure that the Housekeeping Module stops acquiring and telemetering housekeeping data once the dwell sequence stops.

h. Temporarily block telemetry and verify that the Housekeeping Module properly discards data sets, while recording the discards in the Software Housekeeping Data Base. Resume telemetry and ensure that the Housekeeping Module recovers and resumes sending housekeeping data.

i. Send command packets to the Housekeeper Thread and ensure that all commands are acknowledged with an CMD_REPLY_UNKNOWN reply code.

j. Measure the maximum time that the Housekeeper Thread has control. Ensure that this value is less than 32ms.

k. Measure the maximum execution time of the Housekeeper Module's MinorTick Interrupt client function. Ensure that this value is less than 5ms (i.e. less than 20% of the CPU).

l. Verify that the Housekeeper Thread is run only on the ASM EA located in the Side-B Subgroup (i.e. is not run on the ASM EA in subgroup A nor on any of the PCA EAs)

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		