# Graphics Microsystems Inc.

Author : Paul Calinawan

Date : August 1, 1997

# Color Smart (AS98)
## Project

**Abstract:**

This document provides a general overview of the Color Smart System Architecture.

Date Saved 12/18/2017 9:36:00 AM
Date Printed 12/18/2017 9:36:00 AM

## Revision History

| Author | Revision | Date | Remarks |
|---|---|---|---|
| Paul L Calinawan | 00.01 | November 21, 1997 | Initial release |
| Paul L Calinawan | 00.07 | January  4, 1998 | Sent to JimC for<br>Initial review |
| Paul L Calinawan | 00.09 | January 14, 1998 | Sent to JimC for review |
| Paul L Calinawan | 00.25 | February 14, 1998 | Sent to JimC for review |
| | | | |
| | | | |
| | | | |

# 1. WHAT'S NEW (since 00.40)

- ✓ **Added Jim's description of the whole CS Target Align process**
- ✓ **HMM state transitions were revised to improve speed if the Move Settle time is 0**
- ✓ **Measurement Comm is much, more complete**
- ✓ **Took out MaximumBacklash Parameter, not needed by the controller card (will be needed by NT)**
- ✓ **Added memory map diagram for Boot And Main code**
- ✓ **Refined the Document Overview**
- ✓ **Refined the Design Philosophy section**
- ✓ **Timing Parameters, there are now 2 separate move settle times, 1 for Point to Point and another for Target Align move settle. All related diagrams were revised**
- ✓ **Added more "IMPORTANT" notes**
- ✓ **We have new hardware port assignments. Made revisions to reflect these hardware changes**
- ✓ **Added Timing Diagram for Flash and Blink target lamps**
- ✓ **The Heading Numbering errors are now corrected**
- ✓ **Added State Transition diagrams for the BOOT Code**
- ✓ **The LOCK Signal is unused. The HW port table was updated**
- ✓

# 2. Table of Contents

# 3. Document Overview

If viewed on a workstation, this document can be interactive. The table of contents jumps to the section by clicking on the page number. Statements that say "See page No n" are cross-references that also allows you to jump into a given section by simply clicking on the Heading Name or the Page Number.

There are sections with high-level context diagrams. These diagrams show the top-level design of a given module (state machine). These diagrams are normally decomposed to show the lower level details of the design. Sections about Task Description sections describe **what** each tasks will need to accomplish. These sections do **not** describe about **how** they will be accomplished. There are also sections about system interaction, which shows the machines that are involved to perform a given task.

The sections about state transitions show the finer detail about the behavior of a given machine. This document goes hand in hand with the source code. It may be necessary to examine the contents of the Exit Procedures by viewing the source code itself. Each state transition diagram is usually accompanied by paragraph-worth of explanation in order to give an overall idea about the things that happen when the transitions that take place.

The state names used in this document are very representative, although not exact of the state names used in the source code. This can be used as a guide as you browse through the source code itself.

Important notes were added to identify areas that may have ambiguity. These notes may also be rules that the design follows. There is a section near the end of this document that describes the hardware configuration of the Color Smart Controller card.

And finally you will find in the last pages, are notes and diagram conventions as used for this document.

# 4. Color Smart System Overview

## *4.1. Color Smart Controller Card Design Philosophy*

The Color Smart operation is based on the Auto Smart. Many of the functionality of the Auto Smart, which resided in the Host PC, are now implemented within the Color Smart Controller Card. This built-in intelligence puts the Color Smart Controller card within the category of smart cards.

The main goal of the Controller Card is for it to perform all timing critical tasks related to scanning, leaving the NT Application with solely the task of data collection and analysis. In line with this goal, the Controller Card is designed to have enough intelligence to perform coordinated tasks to execute a given procedure. For instance, as a rule, all measurements are done with the Probe Head always lowered. This rule alleviates the NT Application from the need to manage the “Raising and Lowering” of the Probe Head. Again, the aim is to further simplify the operation of the NT Application wherever possible. However, note that in spite of this intelligence, the Controller Card is to avoid any floating-point calculations. Such operations as measurement analysis are left to the NT Application, which runs on a much faster processor.

From the ground up, the Controller Card firmware has been designed using Object Oriented principles. At the same time, the Controller Card also has to meet all of the Color Smart System’s the real time requirements. Once all the real time requirements have been met, speed and size may be sacrificed to facilitate implementation of the Object Oriented design. As an example, the Controller Card firmware uses a Multi-tasking Kernel, which performs the coordination of all the system tasks. This Kernel introduces some timing overhead. However, early analysis shows that the timing overhead is insignificant for the Color Smart’s need. The benefits given by using the Real Time Kernel far outweighs the timing latencies added. Note that there are times where the Kernel is safely bypassed by a process to meet a timing requirement and still maintain an organized running system.

The firmware is written using C++ constructs to ease implementation of the design models (using Virtual Machines). The C++ constructs also impose some timing overhead. But again they are insignificant and Color Smart timing budgets will still be met.

### *4.2. The Real Time Operating System*

The CS Firmware System runs on a proprietary Multi-tasking Kernel designed at GMI.

The tasks are organized into virtual machines. The machines run in a "cooperative manner" where each is responsible for "quickly" giving back the CPU resource.

Event driven.
Message passing.

The Kernel Provides 2 Levels of priorities:

1. High Priority
2. Low Priority

There is no preemption between the 2 levels with the exception of interrupts, which puts the current task on hold while it the interrupt is being serviced.

In general there are 3 threads of operation. All of these tasks are serviced in parallel:

1. Scanning Operations
2. Query Operations
3. LED update Operations

Scanning Operations always uses the High Priority level.

Messages and Events are one and the same.

### *4.3. The Color Smart PCI Card.*

There are 2 separate firmware that run the CS Controller Card.

One is referred to as the **Boot Firmware** and the other as the **Main Firmware**.

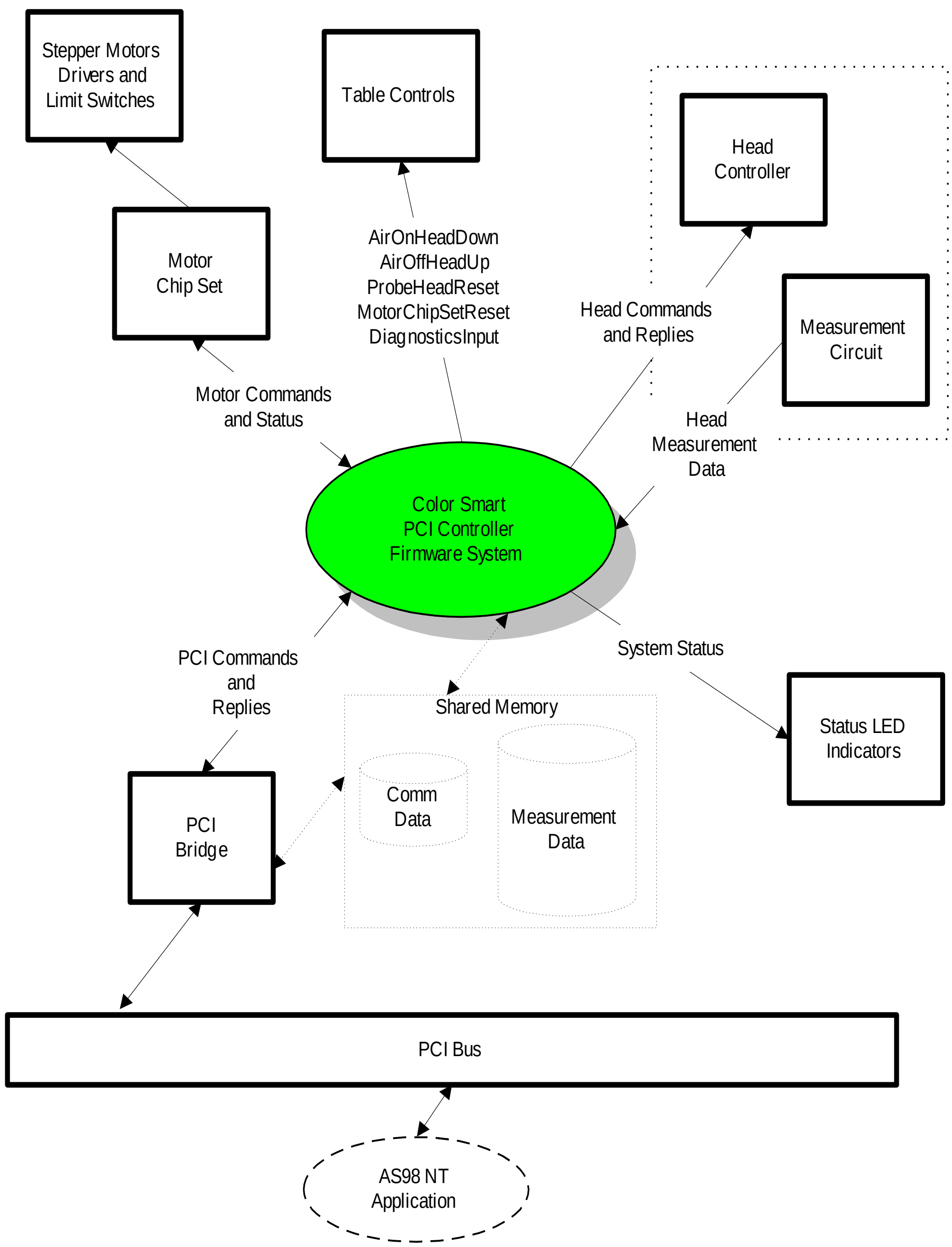
The Boot Firmware resides in a 64K ROM.

The Main Firmware is loaded in the first 256K of shared memory by the NT Application.

386EX CPU
1 MEG RAM
64 K BOOT
PCI BRIDGE - CUSTOM FPGA for GMI
2.1 PCI SPEC

# 5. High Level Hardware to Firmware Interface Diagram

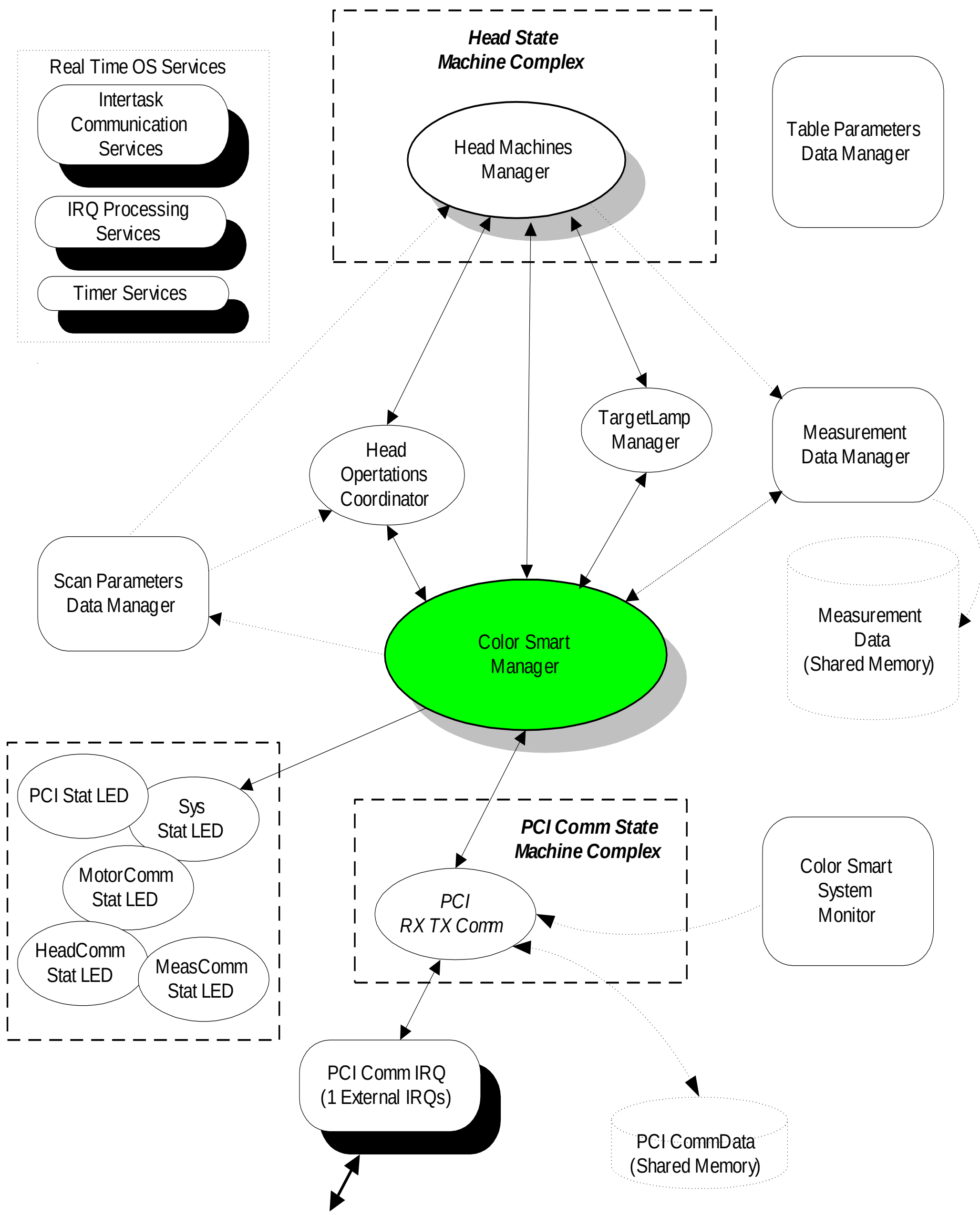
The diagram below shows the relationship of the Color Smart PCI Firmware to the hardware environment providing an overview of how the Color Smart PCI Firmware interfaces to the outside world:



# 6. High Level Firmware Context Diagram

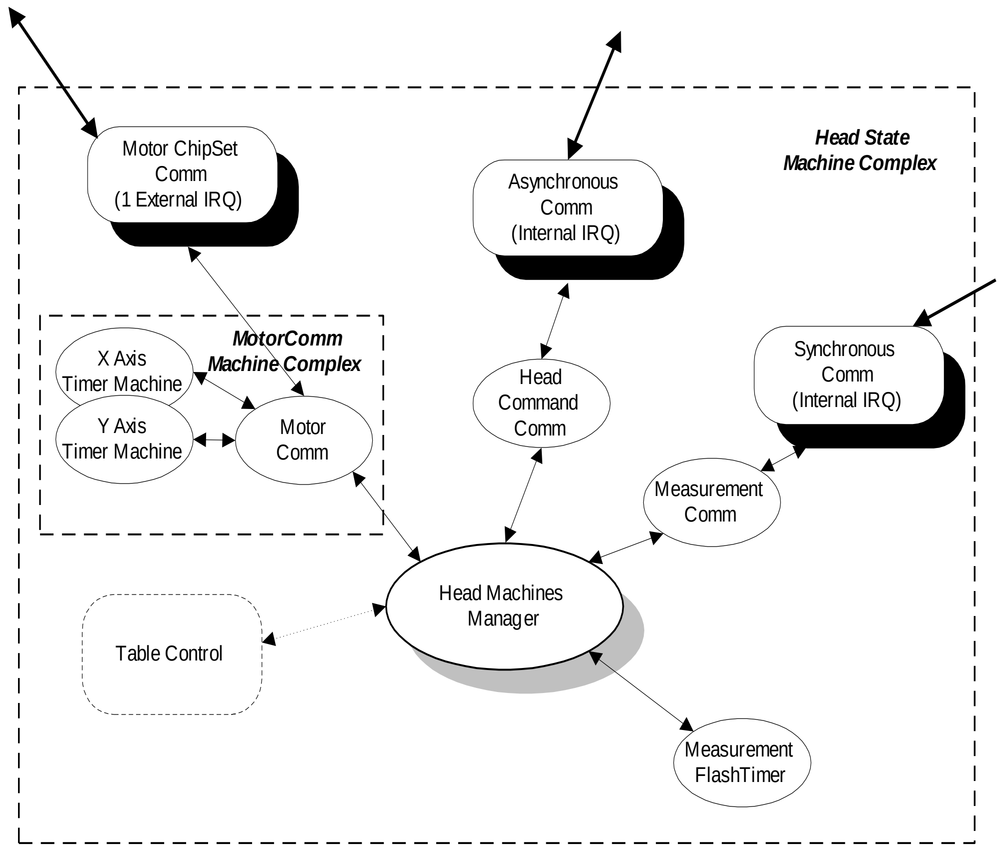
The diagram below shows the different firmware components involved in the operation of the Color Smart PCI system. Some of these components are interfaces to the hardware components as seen in the previous diagram.

Each of these components is further described in detail in the next pages.



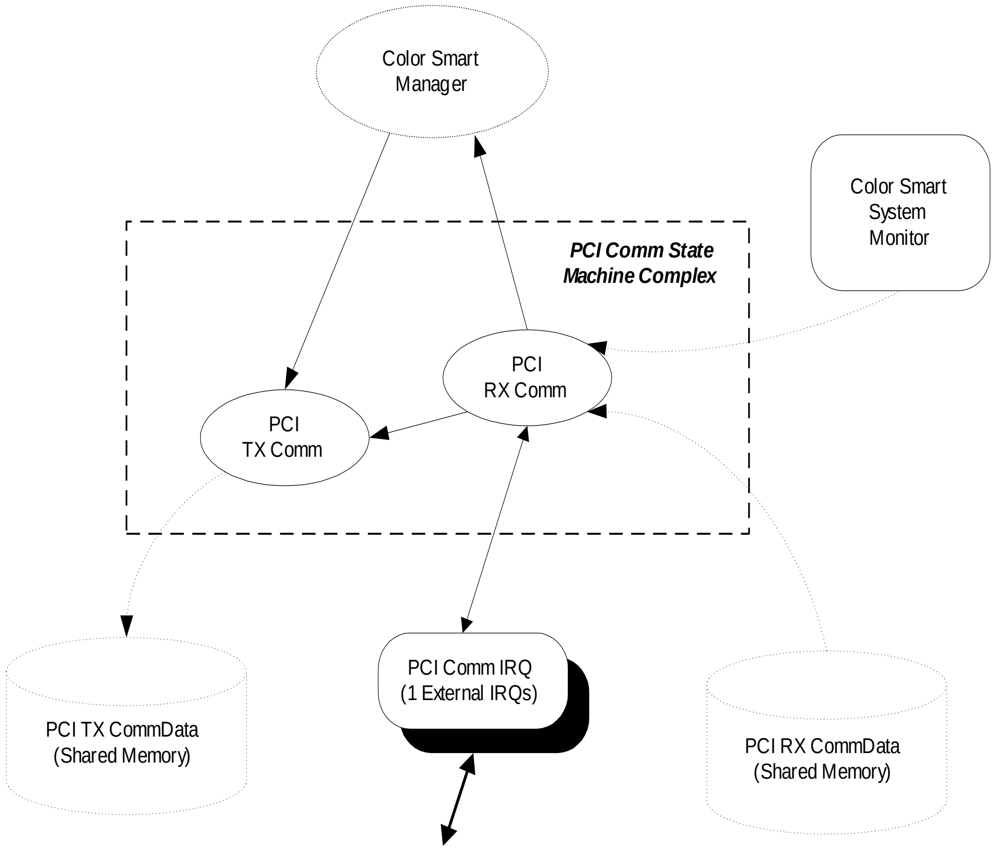
## *6.1. Head State Machine Complex Context Diagram*

The diagram below shows all the machines that reside within the Head State Machine complex. Some of these components interface to the hardware components as seen in the Hardware to Firmware Interface Diagram.



## *6.2. PCI Comm, State Machine Complex Context Diagram*

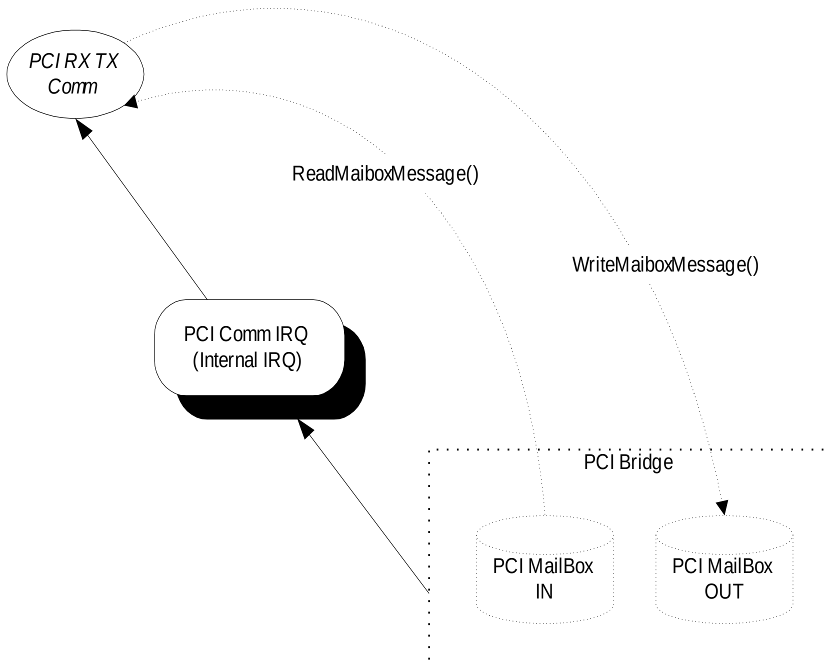
The diagram below shows all the machines that reside within the PCI Comm State Machine complex.



### *6.3. PCI Comm IRQ (IPCM) Task Description*

***IPCM** - **I**nterrupt **P**ci **C**om**M***

This is an interrupt service routine that will handle all the low level communication tasks with the PCI Bridge. Each mailbox is 4 Bytes in size.




NOTE:

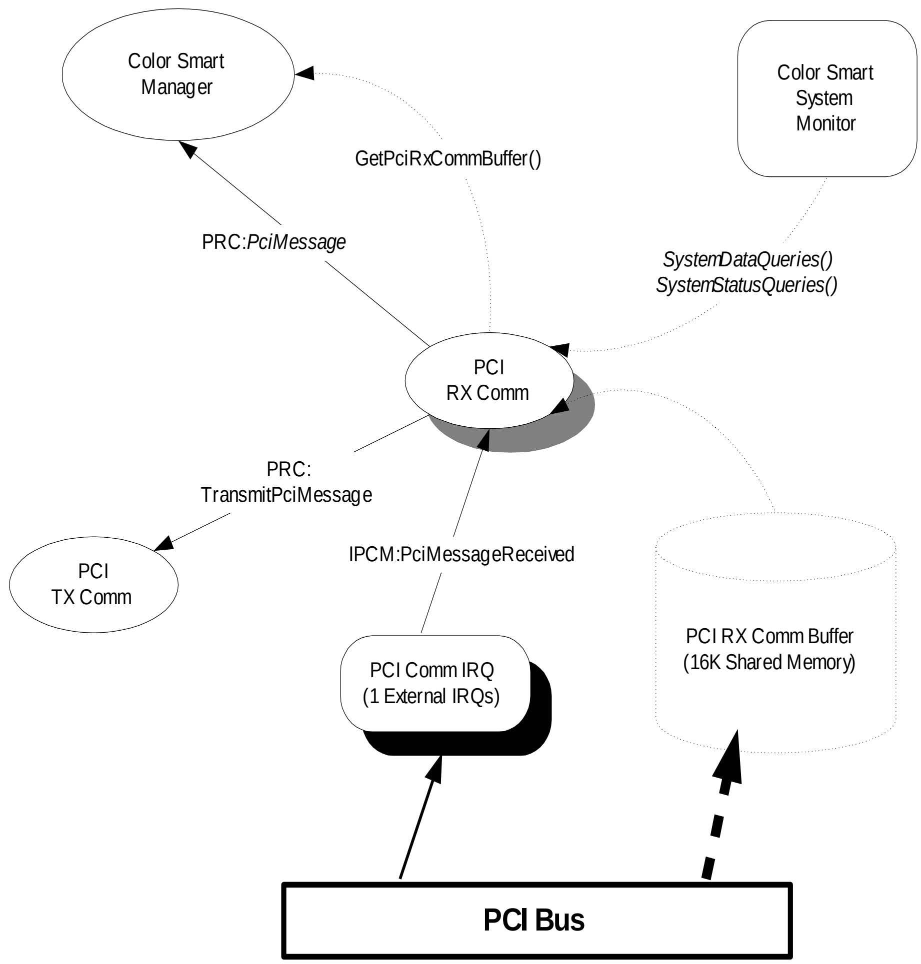
Since the data bus is only 2 Bytes wide, a hardware scheme has been implemented to select either the Low or the High Word of the mailbox location. The table below shows the HW Selection works. See Chip Select Definitions on page 176 for more information.

| Chip Select Enabled | Address Line A1 | Mail Box Selected |
|---|---|---|
| CS4 | HI | Mail Box IN – LOW WORD |
| CS4 | LOW | Mail Box IN – HIGH WORD |
| CS5 | HI | Mail Box OUT – LOW WORD |
| CS5 | LOW | Mail Box OUT – HIGH WORD |

### 6.3.1.1. PCI RX Comm (PCM) Task Description

***PRC – P****ci* ***R****x* ***C****omm*

The PCI Rx Comm state machine (SM) will handle all the communication that is transacted between Color Smart NT Application and the Color Smart PCI Card. This communication is based on the NT - CS Controller Exchange Protocol. NT Messages are processed and analyzed by PRC. He then makes a translation and sends an equivalent *PciMessage* that can be understood by CSM.




NOTE:

Due to the “Full Duplex” nature of the PCI communication protocol, there are 2 machines designed to handle each of the communication buffers.

### 6.3.2. PCI Comm Receive Buffer

The PCI Comm Receive buffer is a shared resource between the NT Host and the CS Controller Card. If the NT message needs to have data, then this data will be found in this buffer. Parameter setting for “Point Measurement Coordinates” is one example on when this buffer is used.

The size of the PCI Comm Receive Buffer was chosen to accommodate 1000 scan coordinate data. Each coordinate data is 10 Bytes:

1000 x 10 Bytes = 10 K

**IMPORTANT: It is implied that the Controller Card will never write in this section of memory. See memory mapping section.**

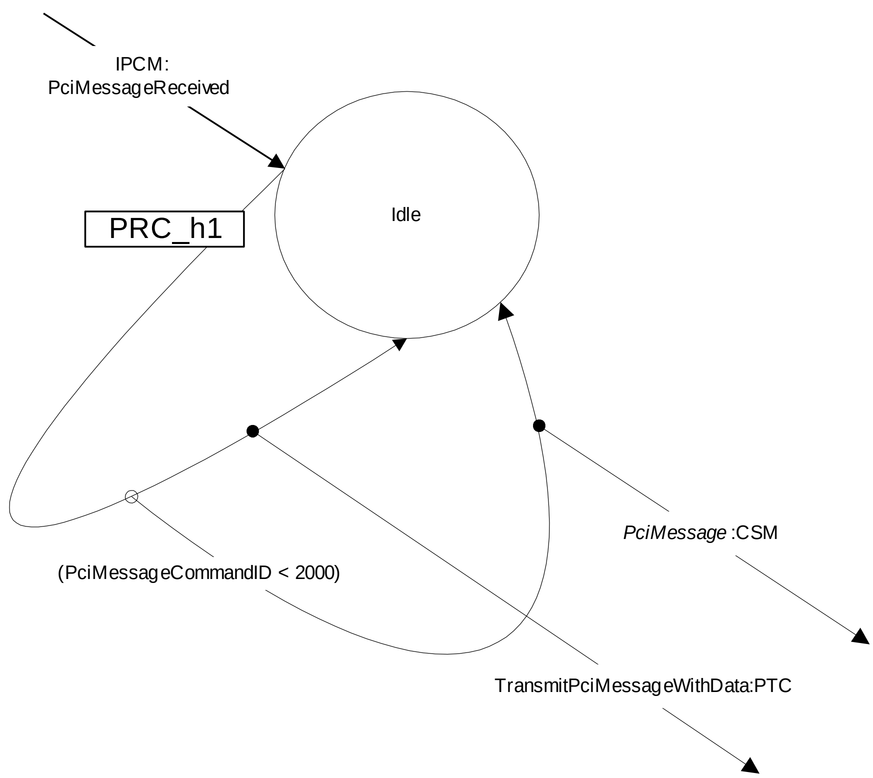
**The NT Host should not perform another write to this buffer unless it has received an ACK message for the previous command.**

### 6.3.2.1. PCM – Pci Message Received, State Transitions

The “PciMessageReceived” message is generated by IPCM.

The diagram below shows that PCM is relatively simple. To add to its simplicity PCM only has one state. When it gets a “Pci Message received” event, it examines the message that is currently in the mailbox to determine whether it is a query message or a message that requires a system operation.

NT messages that are not status or data queries requires the attention of CSM and are directed appropriately to him. Otherwise a reply with data is made to service the query request.



### 6.3.2.2. PCI TX Comm (PCM) Task Description

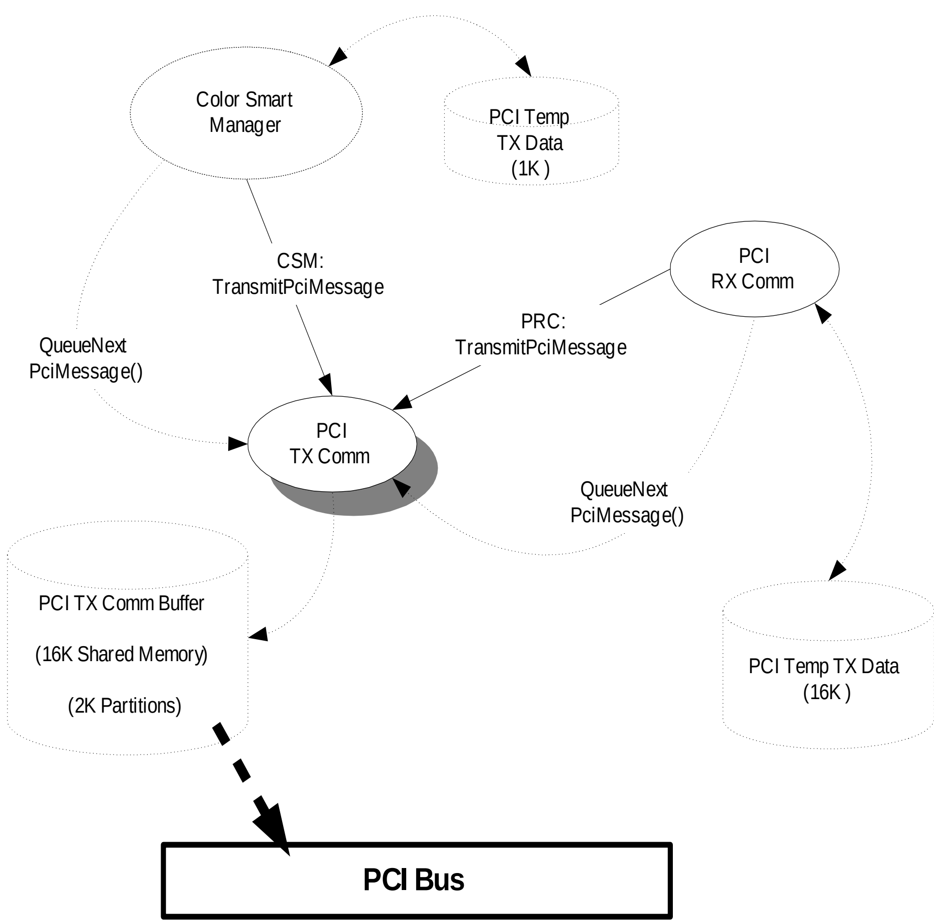
***PTC – P****ci* ***T****x* ***C****omm*

The task will need Timer where the timing services of the OS Kernel will be utilized. The Color Smart NT Application will be able to communicate with the Color Smart PCI at any point of the Color Smart PCI operation.

The PCI Communication Protocol requirements allow for multiple replies for a given PCI Message. For this, PCM needs to have the ability to queue outgoing messages to the NT Host. Queued messages are sent a soon as the NT Host has accepted (read) the last message. The message queue as well as the “PCI Transmit Ready Flag” will be checked at a given timing interval.

### 6.3.2.3. PTC – System Interaction

As the diagram shows, PTC interacts with 2 other machines, CSM and PRC. There are 2 steps involved for the machines to send a Pci message. First they have to “Queue” the next Pci Message and secondly a TransmitPciMessage is sent. The first action actually stuffs the Pci Message and Data (if available) and second informs PTC that a new data has been added. PTC takes care of the rest.



### 6.3.3. PCI Comm Transmit Buffer

The PCI Comm Transmit Buffer has a special structure. Its total size is 16 K. Due to the PCI Communication protocol requirement, it has been designed to provide 8 PCI message Cells in 2 K partitions. This queue is configured as a circular buffer.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Cell #1 | Cell #2 | Cell #3 | Cell #4 | Cell #5 | Cell #6 | Cell #7 | Cell #8 |

Each Cell contains a header as follows:

| **Length** | **Command And Extension** | **Param1** | **Param2** |
|---|---|---|---|
| WORD | WORD | BYTE | BYTE |

**Length** is the size of the PCI Message (in BYTES)

The Following data on the header is placed in the outgoing mailbox to generate the interrupt request to NT. The following are described within the PCI Exchange protocol document:

1. **Command and Extension**
2. **Param1**
3. **Param2**

**IMPORTANT:**

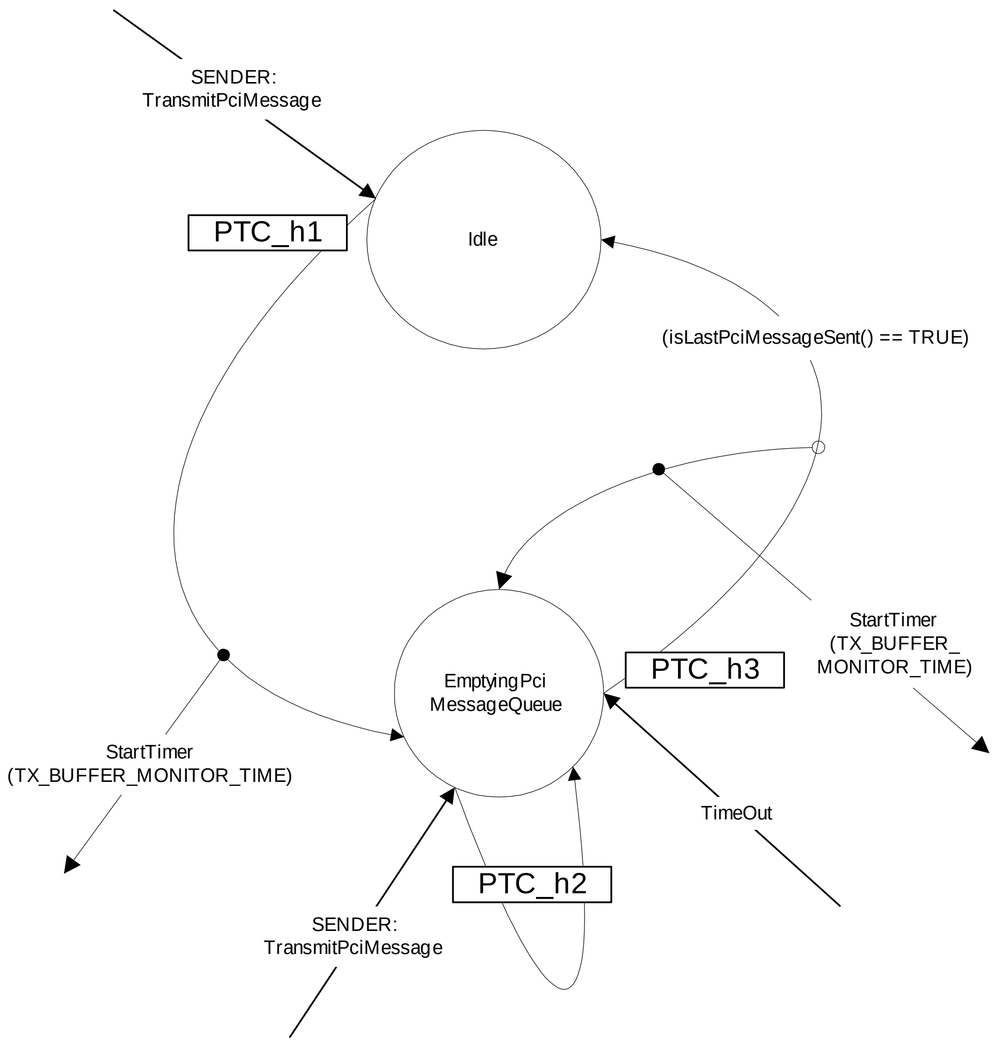
**Theoretical Worst case timing where the circular buffers might be overwritten.**

### 6.3.3.1.1. PCM - Transmit Pci Message, State Transitions

Either CSM or PRC generates the “TransmitPciMessage” message.

NOTE:

The idle state implies that the “Mailbox is Ready” and that there are no Pci Messages in the queue.



## 6.4. Color Smart Manager (CSM) Task Description

### ***CSM - C****olor* ***S****mart* ***M****anager*

This SM has the task of coordinating the overall operation of the Color Smart PCI Card:

- He knows the overall condition of the system
- He knows the command set specification
- He is responsible for generating the replies to the commands given by the Color Smart NT App
- He decides when a request can be accepted or denied
- He makes sure that the message data is within their valid ranges before being passed to the rest of the system.
- The NT requests are processed based on the current CSM mode
- The CSM changes from one mode to another again upon the request of NT
- He relays and receives messages from the Head Coordinator as well as to the Scan coordinator
- He relays the system condition to the State LED SM
- CSM is in charge of putting the head back in resting place (parked at white plaque) when it is in the “At Home Idle” state
- He is in charge of making sure that the Target Lamp is turned off when it times out at any state of the system operation
- In general, the CSM states are representative of what the system as a whole is currently doing
- He is the boss.

### 6.4.1. CSM Modes of Operation

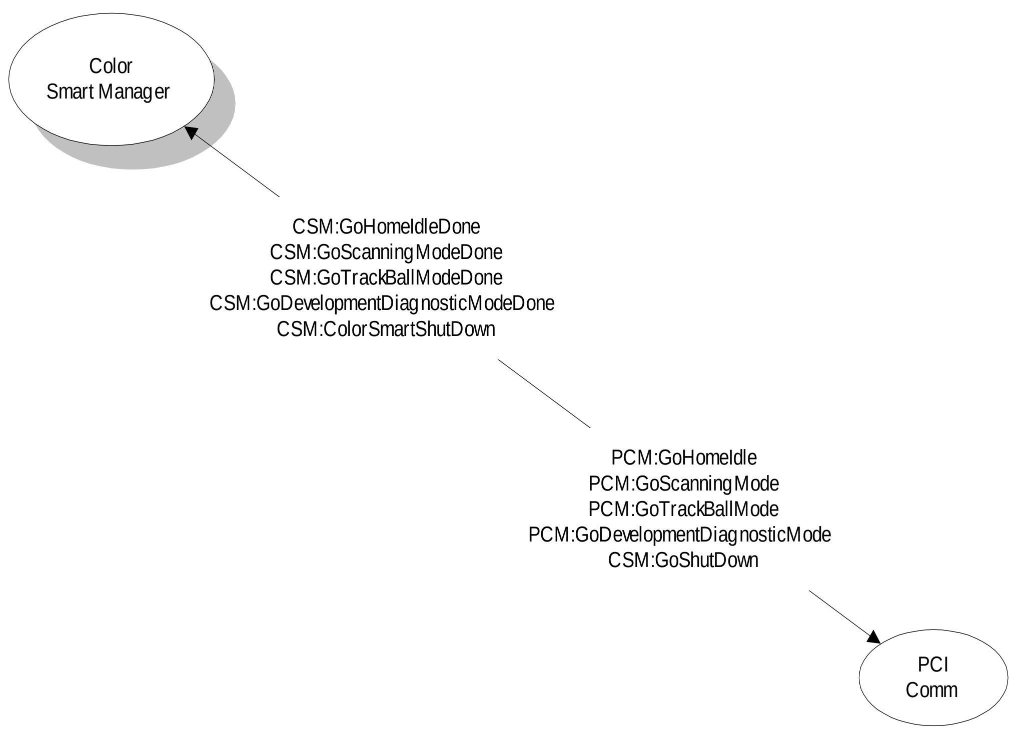
CSM or the Color Smart in general, operates in one of the 5 modes of operation. These are:

1. **SHUT DOWN Mode**
2. **IDLE AT HOME Mode**
3. **SCANNING Mode**
4. **TRACKBALL Mode**
5. **DEVELOPMENT DIAGNOSTIC Mode**

The different modes allow the Controller Card to determine how the system should behave. This includes the automatic setting of the Air Bearing the automatic selection of move profiles among other things.

**IMPORTANT: These modes only pertain to the CS Controller Card operation. They are NOT synonymous to the modes of the Color Smart table itself.**

#### 6.4.1.1. CSM – System Interaction, *Mode Change Requests*

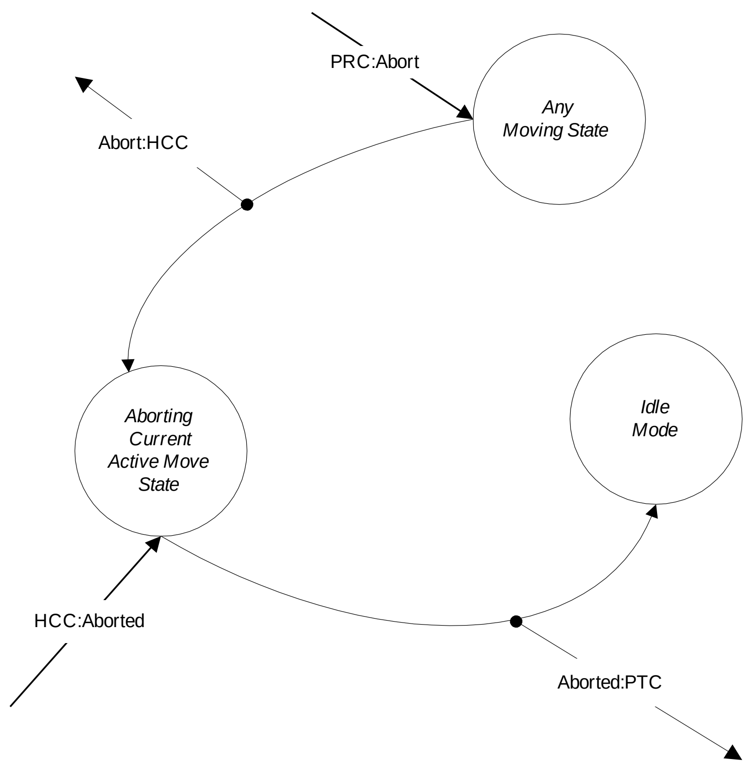


### 6.4.1.1.1. CSM – Abort, State Transitions

The “Abort” message is handled at any state where the probe head might be actively moving. The message forces CSM to go back into its own Idle State. These modes are:

1. Scanning Mode - Idle
2. Track Ball Mode - Idle
3. Development Diagnostic Mode -Idle

An “Aborted” message is replied back to PCM who then relays it to the NT Application. The abort procedure simply stops the head from moving, if it is moving. It is up to the NT Application what to do next.



### 6.4.1.1.2. CSM – Initial, State Transitions

There are a few things that are required for the NT Application to perform before CSM gets out of the “UN-initialized State”. Several parameter downloads are needed before you can move on to the next state. The table attributes, the move profiles as well as the timing parameters have to be known before any kind of move request can be processed and executed.

Note: The parameter downloads may be done in any order.

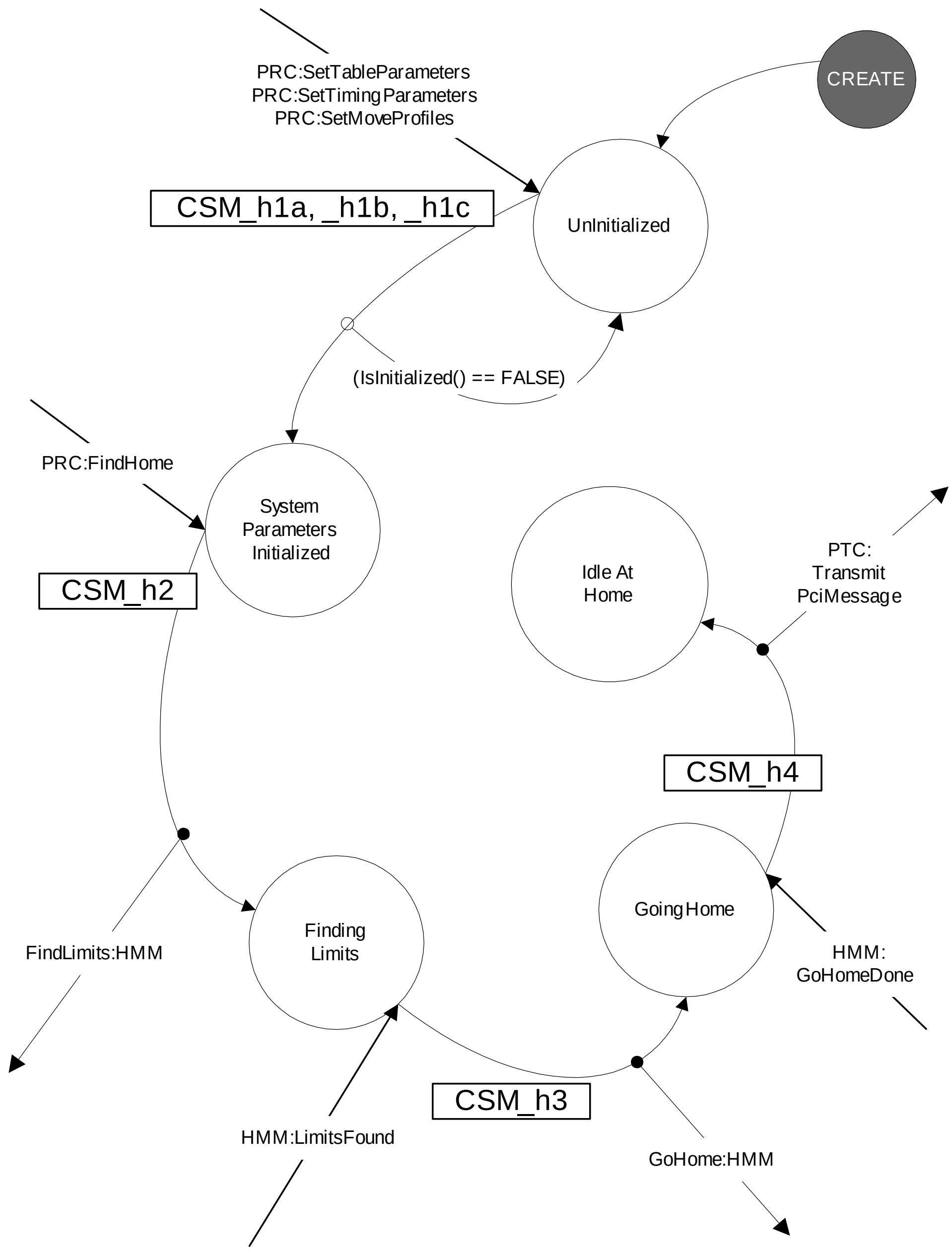
Once CSM has moved from the UN-initialized State into the “Parameters Initialized” state, a Find Home command needs to be given by the NT Host before CSM can proceed to the Idle At Home State and later to other states for normal move operations.

Before the NT Application exits, a GoShutDown message should be sent to the controller card. Motor Power will be disabled while in the Shut down Mode. Note that the Shut Down message can only be sent within the 2 following CSM modes:

1. IDLE AT HOME Mode
2. DEVELOPMENT DIAGNOSTIC Mode

The next diagram shows all of these initial state transitions that need to take place for CSM to finally settle in the “Idle At Home” state.

When CSM receives a Find Home message while it is in the Shut Down State, he will perform a “Find Limits” as well as a “Go Home” sequence. The completion of the request puts CSM in the Idle At Home State or mode.




Note:

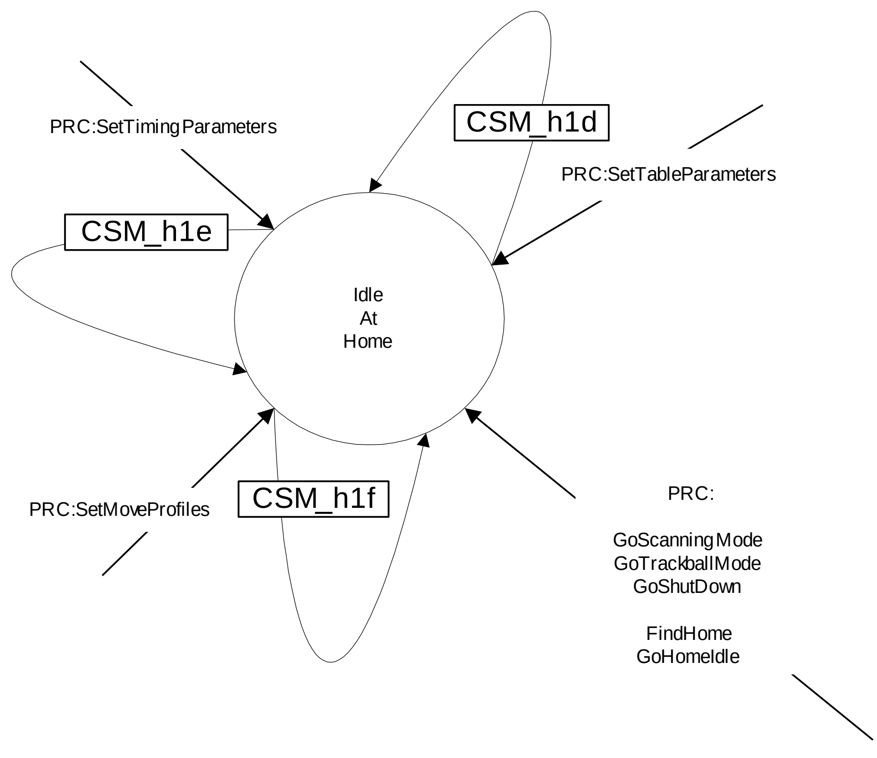
When CSM is in the Home Idle State, the air is off and the measurement sensor is at the middle of the white plaque. This is considered the “parked” position.

### 6.4.1.1.3. CSM – Idle At Home, Messages Handled

The Idle At Home State is simply a starting point for all CS operation. This state implies that the probe head is sitting at the Home position (coordinate x=0, y=0), and that the Air Bearing is off. Any transitions from this state require a full calibration to be executed. This behavior is seen in the next pages.

The diagram below shows all the messages that CSM will handle while it is in the Idle At Home State. The next pages will show how CSM handles each of these messages as well as the state transitions that occur based on the event.

Note: Parameter settings may once again be changed while in this idle state and At Home mode.



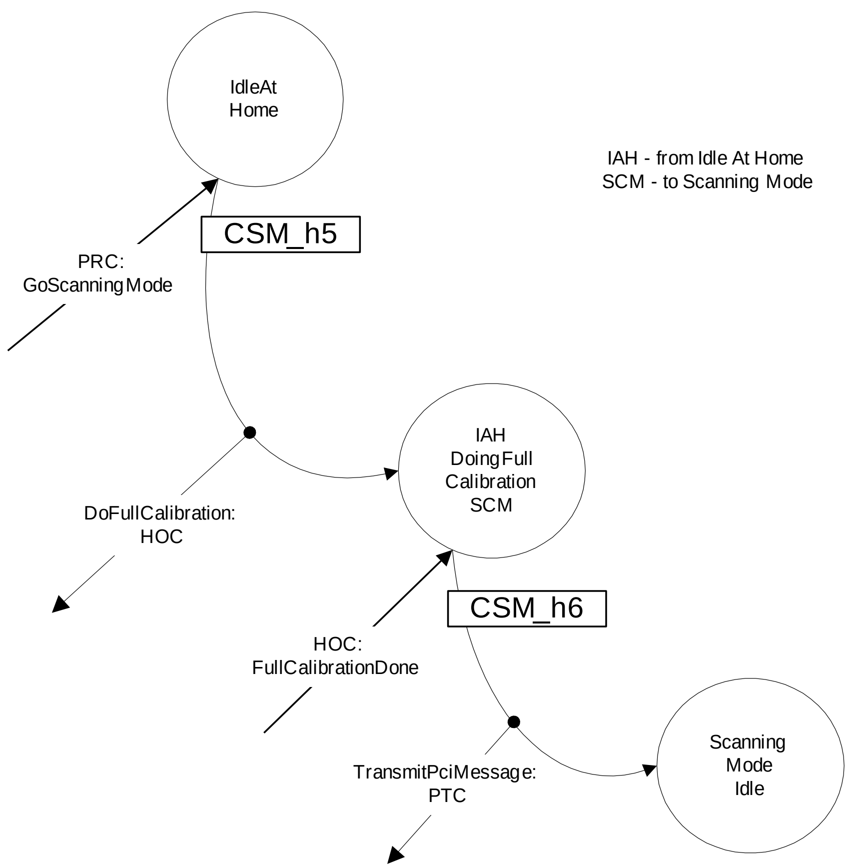
### 6.4.1.1.4. CSM (Idle At Home) – Go Scanning Mode, State Transitions

When CSM receives the "GoScanningMode" request while he is in idle, he makes a request to HOC to perform a full calibration procedure and waits until HOC has completed the request. He then informs the NT Application that he is in the Scanning mode, ready for the next operation.

A full calibration consists of the following:

1. Limit Reference re-calibration
2. Black Calibrations
3. White Calibrations

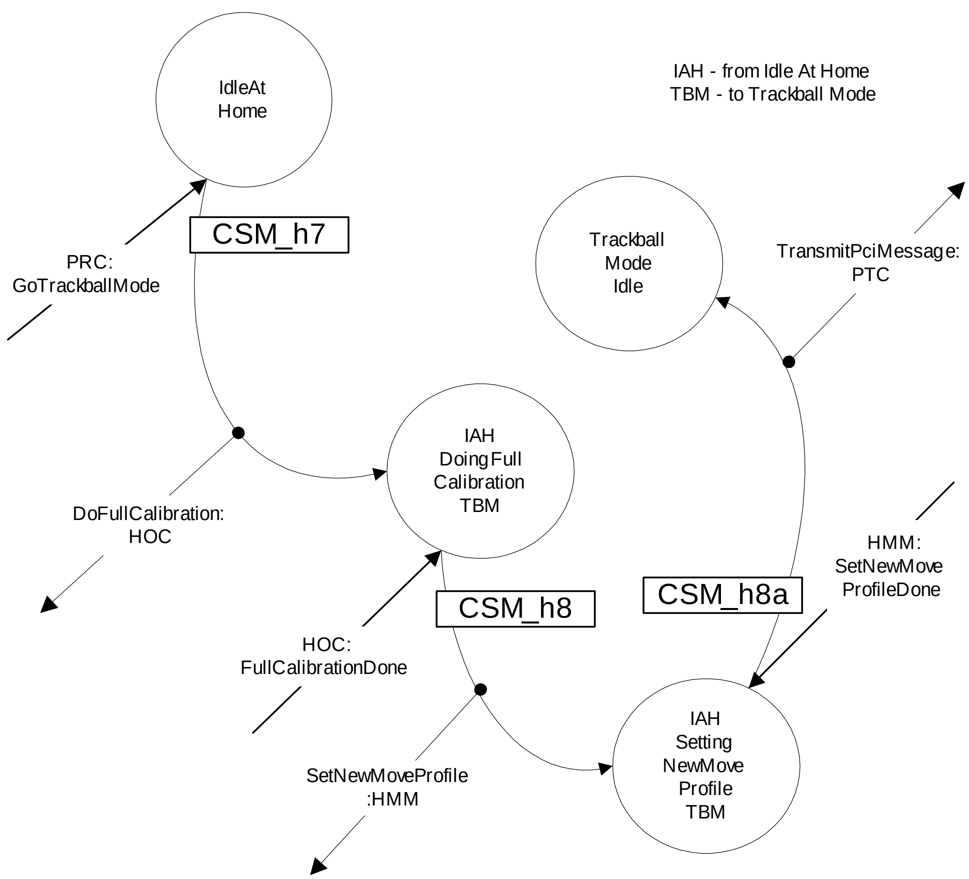
See on page for more information.



### 6.4.1.1.5. CSM (Idle At Home) – Go Track Ball Mode, State Transitions

This procedure is basically the same as the Go Scanning mode except that it places CSM in the Trackball Mode when the process is complete. Again, a full calibration procedure is performed and the NT Application is informed when CSM is in the Trackball Mode and ready.

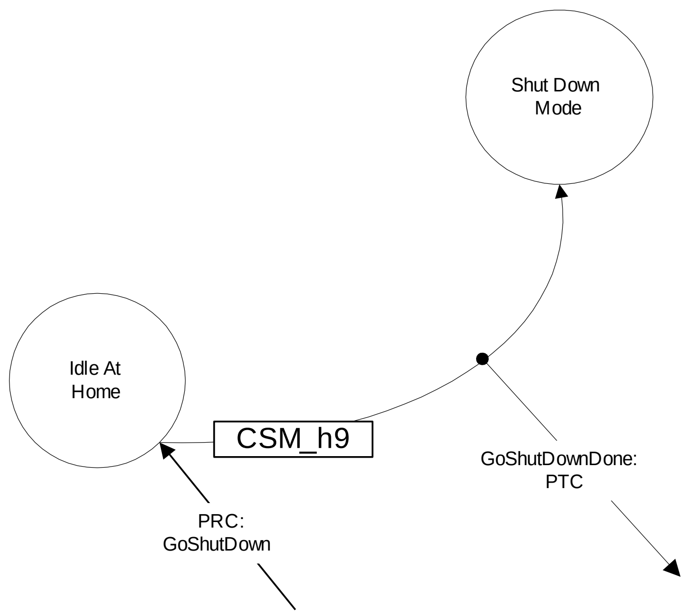
Note: The move profile is updated to the default Trackball Profile.



### 6.4.1.1.6. CSM (Idle At Home) – Go Shut Down, State Transitions

Shut Down request is made just before the end user intends to close the NT Application and power down the host computer. Operational requests are no longer accepted or processed while in the "Shut Down Mode".

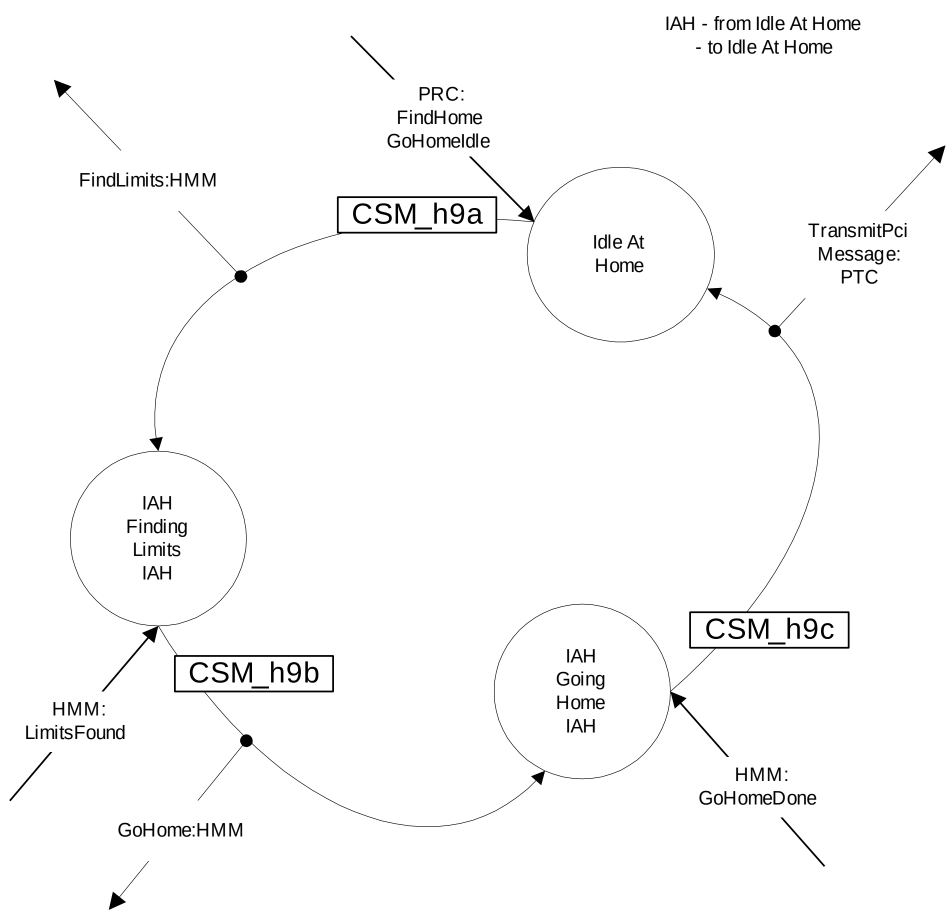
Note: Query messages may be processed by PRC since he is independent of CSM.



### 6.4.1.1.7. CSM (Idle At Home) – Find Home, State Transitions

This request forces CSM to re-find the limits even though it is already in the At Home Idle State. This can be used during the "Find Home – Factory Calibration" process where a reference point on the paper guide (gripper guide) is located.

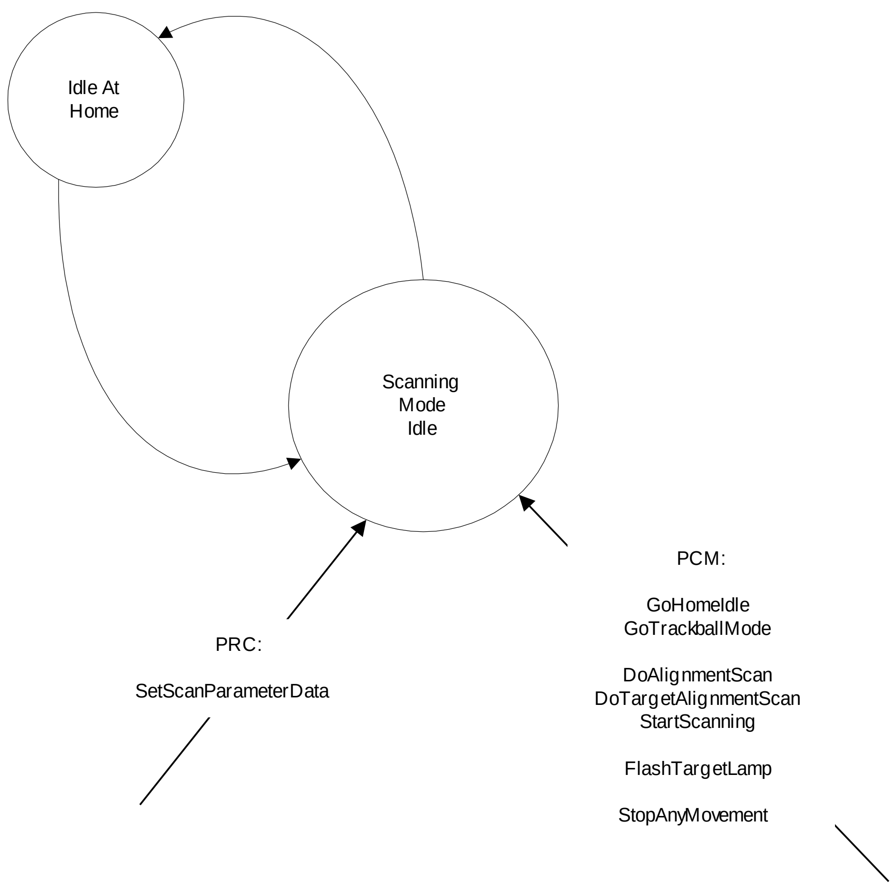
Note: When a "Go Home Idle" request is made when CSM is already at Home Idle, CSM simply reinitiates a Find Limits procedure. The NT application may make this request when the Adjust Home Position process is finished to reposition the probe head in a new home location.



### 6.4.1.1.8. CSM – Scanning Mode Idle, Messages Handled

CSM only accepts a particular set of commands while in the Scanning Idle Mode. The Idle State only indicates that the Probe Head is raised and the Air bearing is off. The Idle State does not indicate the location of the Probe Head.

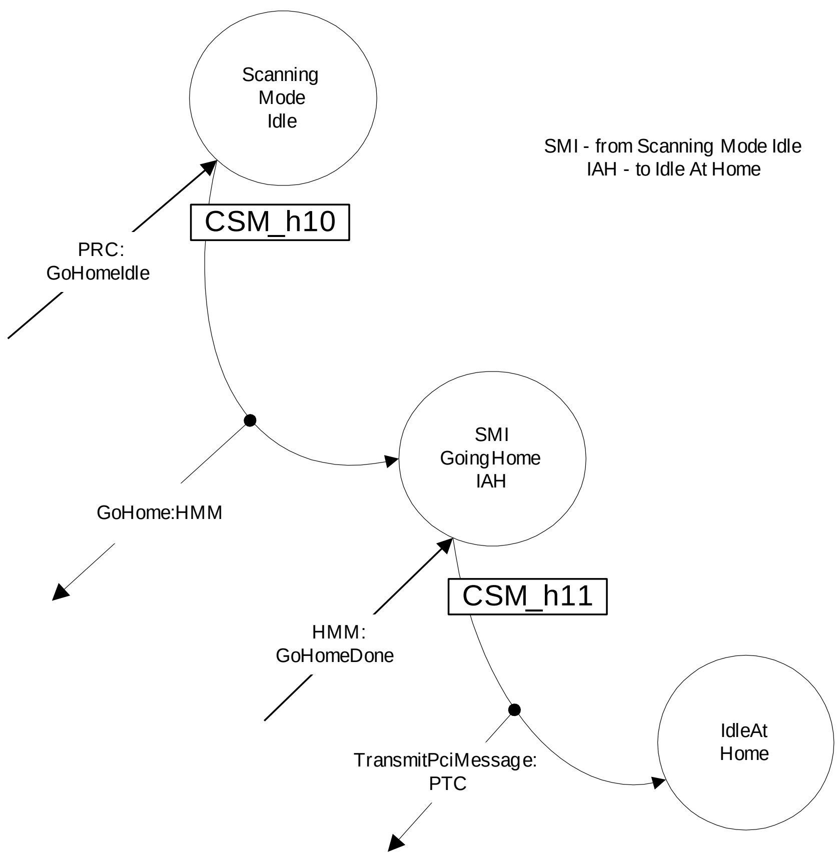
The diagram below shows all the messages that CSM will handle while it is in the Scanning Mode Idle State. The next pages will show how CSM handles each of these messages as well as the state transitions that occur based on the event.



### 6.4.1.1.9. CSM (Scanning Mode Idle) – Go Home Idle, State Transitions

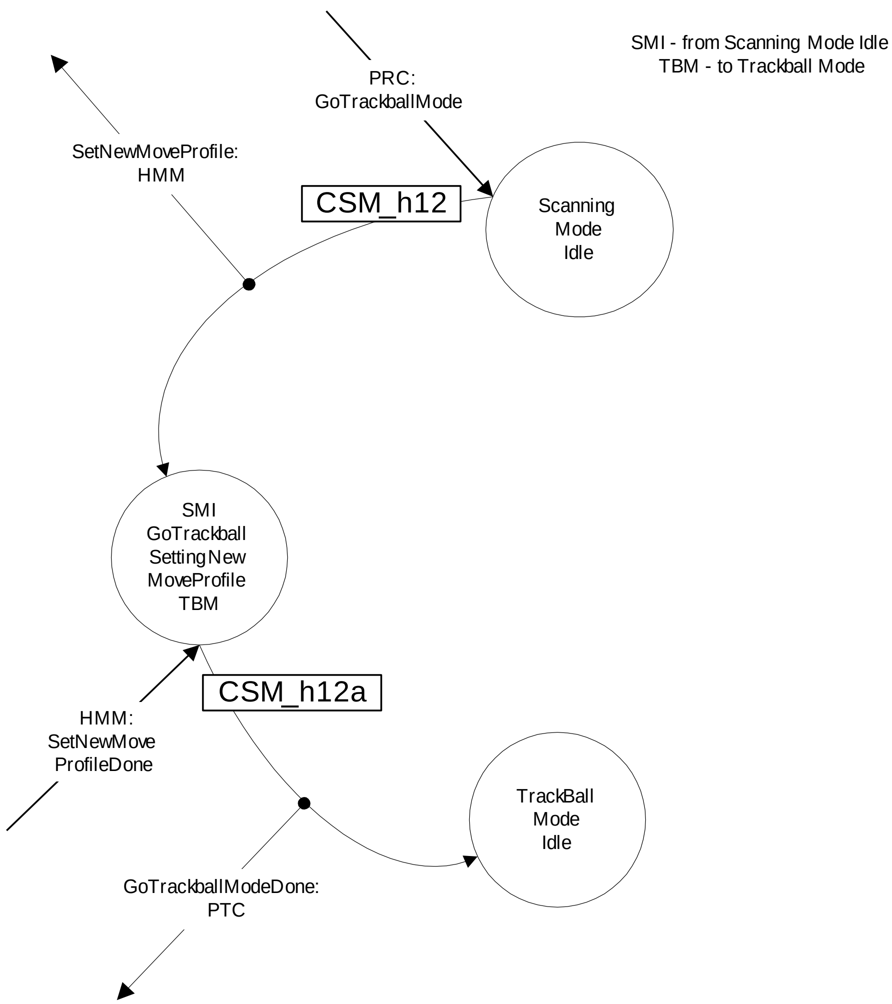
The NT Application normally makes this request when scanning job is complete. CSM sends a direct request for HMM to “Go Home”. HMM informs CSM when he has completed with the request. The NT Application is finally informed when the whole process is complete.

Note: This request also normally follows an Abort sequence.



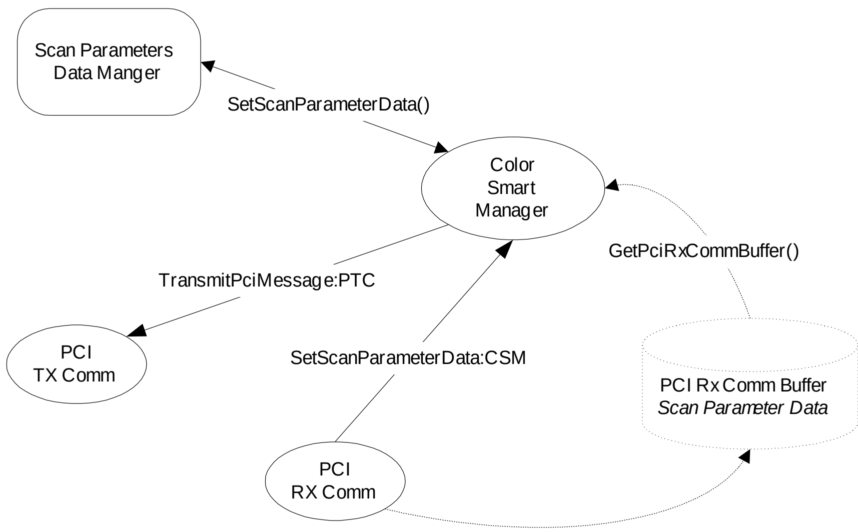
### 6.4.1.1.10. CSM (Scanning Mode Idle) – Go Track Ball Mode, State Transitions

This request allows switching the mode from Scanning Mode to Track Ball Mode. This can be used during the “Add Alignment Point” process where the user uses the Trackball to indicate the next alignment point after one has just been added.



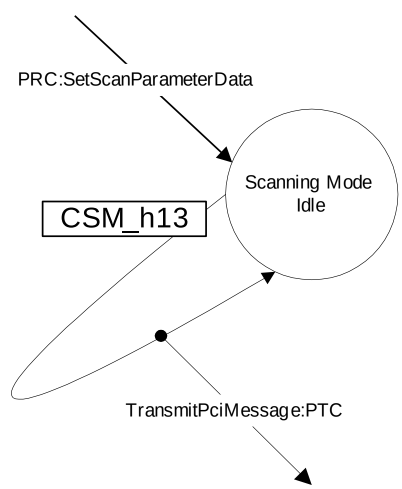
### 6.4.1.1.11. CSM – System Interaction, Scanning Mode Idle, Set Scan Parameter Data

The diagram below shows the machines involved in the process of setting the Scan parameter data.




### 6.4.1.1.12. CSM – (Scanning Mode Idle), Set Scan Parameter Data, State Transitions

This message tells CSM that the scan parameter data is available. CSM gives the scan parameter data that is currently in the PCI Comm buffer over to the Scan Parameter Data Manager (SPDM) for storage. The NT Application is informed when the process is complete. See



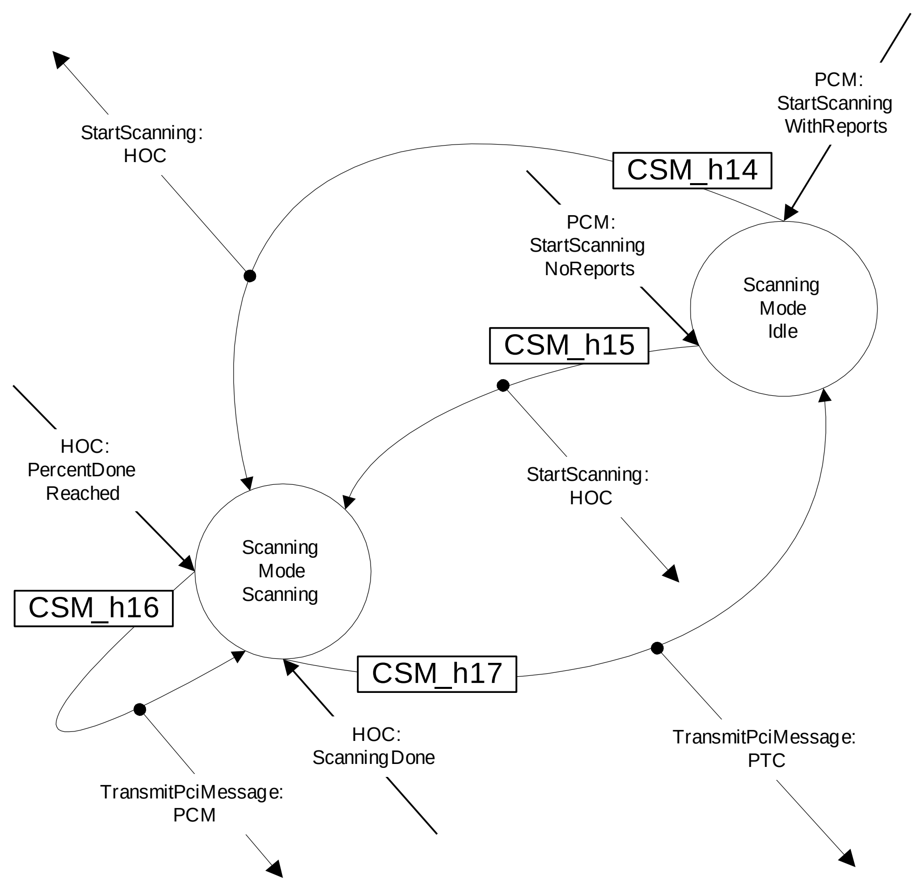
### 6.4.1.1.13. CSM (Scanning Mode Idle) – Start Scanning No Reports and With Reports, State Transitions

The Start Scanning message is normally sent right after the Scan Parameters Data have been set. There are 2 types of Start Scanning messages:

**With Reports** - CSM is required to report when ever a percentage of point measurements have been reached. A final report is also made for the completion of the whole scan.

**No Reports** – CSM is not required to make any reports except for the completion of the whole scan.

HOC is responsible for keeping track of the number of measurements completed. See on page for more information. The diagram below shows that any of the 2 Start Scanning messages may be processed while in the Scanning Idle mode. While scanning, CSM may accept a Percent Done Reached message in which case it informs the NT Application about the partial completion. The "Scanning Done" message signals the completion of the whole scan. Again the NT Application is notified.



### 6.4.1.1.14. CSM – System Interaction, Scanning Mode Idle, Do Alignment Scan

The NT Application issues two of these request to determine the location of the alignment point. Each request will have a different set of alignment parameter data, on for the X direction alignment and another for the y direction alignment. The NT Application performs some analysis of the first scan to determine how the second should be performed.

See HMM – Alignment Scans on page139 for more information.

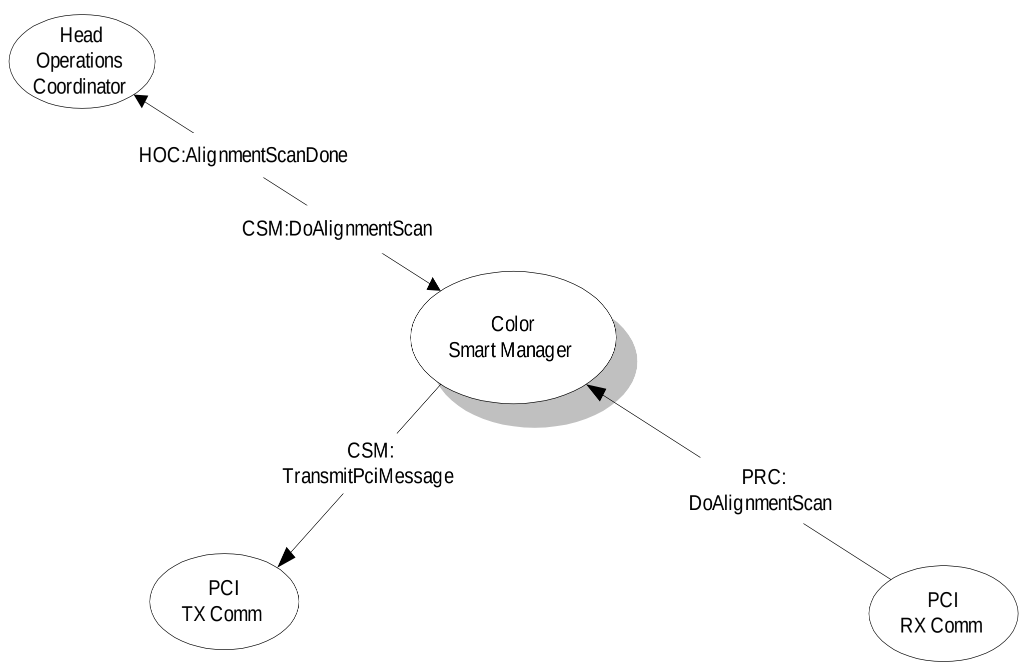
**IMPORTANT: This process assumes that the scan will either be performed in the X direction or the Y direction. This implies that the 2 sets of point coordinates will either have an equivalent X or Y value.**

**X1 = 100, Y1 = 555**
**X2 = 100, Y2 = 888 or**

**X1 = 333, Y1 = 200**
**X2 = 777, Y2 = 200**

**CSM will reject the request if this rule is not followed.**

Below is a diagram that shows the machines involved in the process.

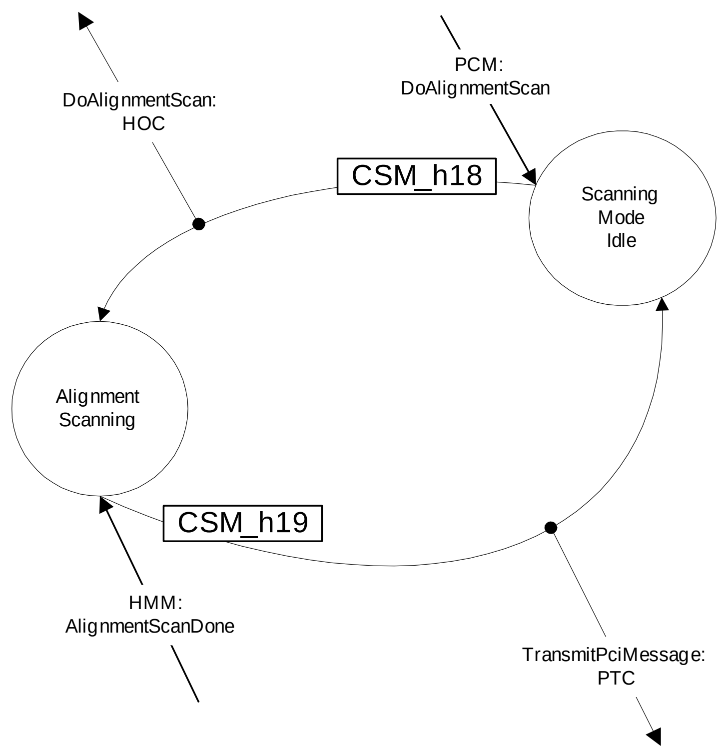


### 6.4.1.1.15. CSM – (Scanning Mode Idle), Do Alignment Scan, State Transitions

The Do Alignment Scan message is an NT Application request that is accompanied by data. This data is stored in the PCI Communication Buffers. The message relayed by PRC is also accompanied with data as follows.

Data 1 = 0 – Flag, Raise Probe Head / Air Off when done
1 – Leave Probe Head Down / Air On when done

Data 2 = Measurement are taken with this given Step Intervals




Note:

Before the Alignment Scan procedures are executed, the NT Application may perform a "Classify Alignment Bar" procedure. This procedure allows the NT Application to determine the orientation of the alignment bar (point). To perform the Classify procedure, the NT Application will simply send a set of point to point measurements command just like a regular point to point scan process.

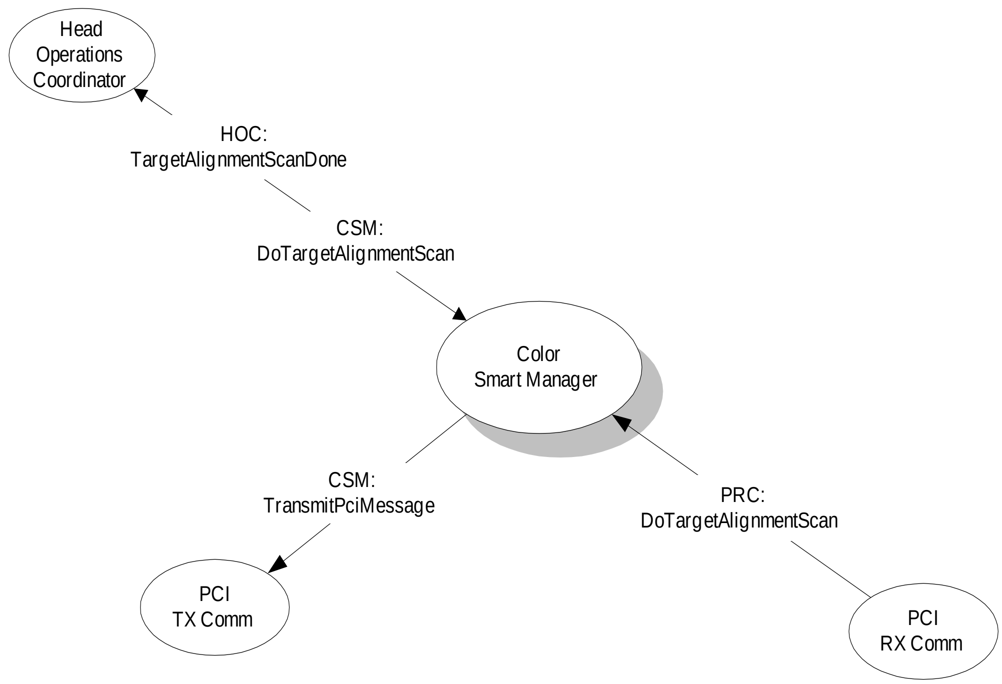
### 6.4.1.1.16. CSM – System Interaction, Scanning Mode Idle, Do Target Alignment Scan

The NT Application issues two of these request to determine the location of the alignment point. Each request will have a different set of alignment parameter data, on for the X direction alignment and another for the y direction alignment. The NT Application performs some analysis of the first scan to determine how the second should be performed.

See HMM – Alignment Scans on page139 for more information.

**IMPORTANT: The same rule noted for the Alignment Scan also applies for this process. CSM will reject the request if this rule is not followed.**

Below is a diagram that shows the machines involved in the process.

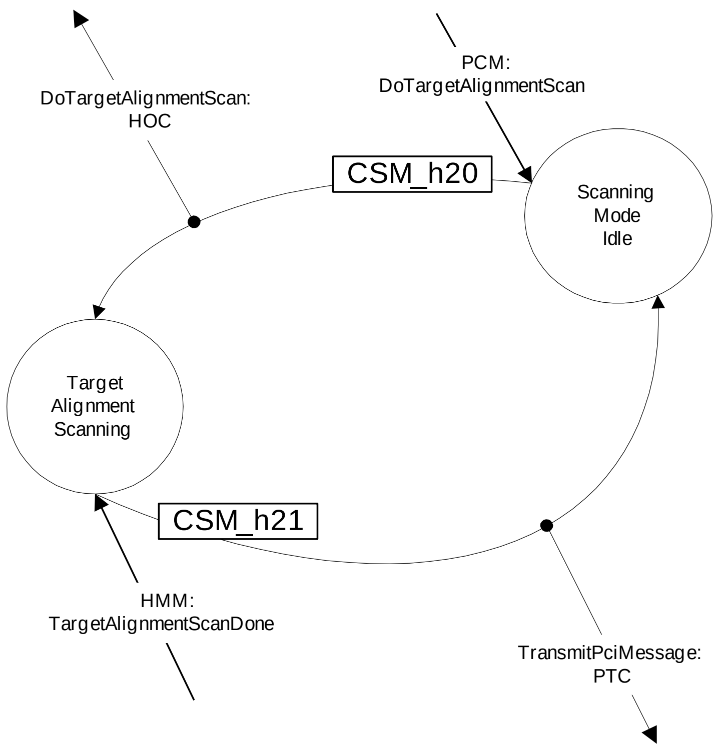


### 6.4.1.1.17. CSM – (Scanning Mode Idle), Do Target Alignment Scan, State Transitions

The Do Target Alignment Scan message is an NT Application request that is accompanied by data. This data is stored in the PCI Communication Buffers. The message relayed by PRC is also accompanied with data as follows.

Data 1 = 0 – Flag, Raise Probe Head / Air Off when done
1 – Leave Probe Head Down / Air On when done

Data 2 = Unused

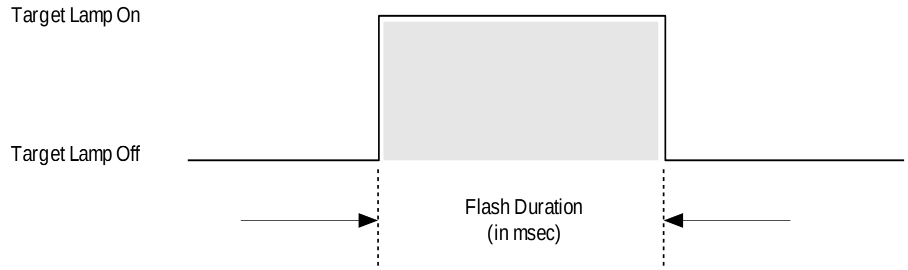


### 6.4.1.1.18. CSM – System Interaction, Scanning Mode, Flash Target Lamp

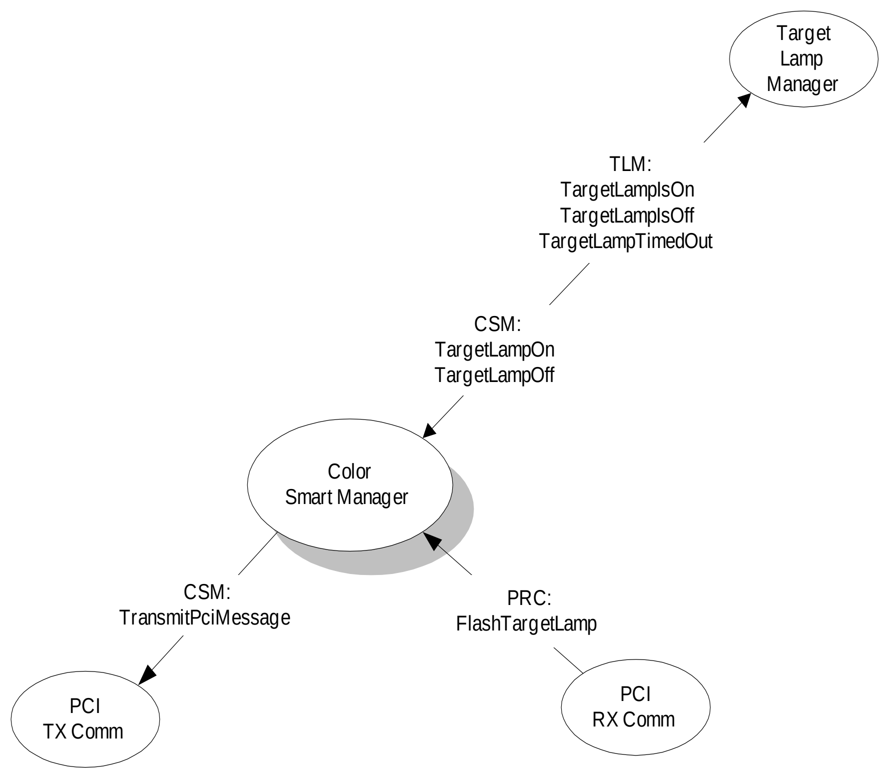
The Flash Target Lamp is normally used by the NT Application to acknowledge that a point alignment has been successfully located by the Color Smart system. While in the Scanning Mode Idle State, the Target Lamp is turned off.

Note: The NT Application will have to wait until the "Target Lamp Flash" process is complete before it sends the next operational request – when CSM is back in the Scanning Mode Idle State.

Flash Target Lamp Timing Diagram:




Below is a diagram that shows the machines that are involved in this process.



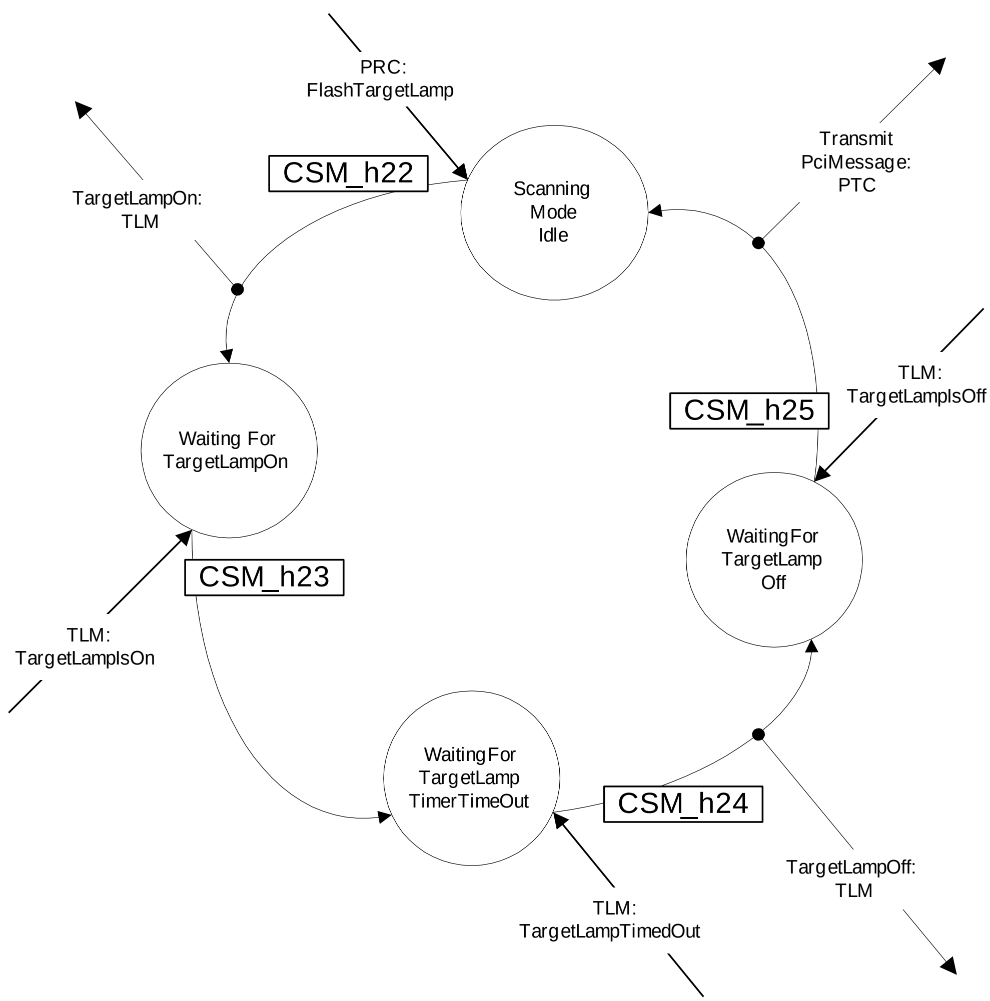
### 6.4.1.1.19. CSM – (Scanning Mode Idle), Flash Target Lamp, State Transitions

When a "Flash Target Lamp" request is received, CSM in turn makes a request to TLM to turn on the lamp for a given duration - *"Flash".* CSM is notified when the Target Lamp timer expires. A Target Lamp Off request is then made. The NT Application is finally informed that the request is complete when TLM says the Target Lamp is off.

The Flash Target Lamp message is accompanied by data:

Data 1 = contain how long the target lamp stays on (in 100 msec units). Note: The max Target Lamp limits this parameter.

Data 2 = Unused



### 6.4.1.1.20. CSM (Scanning Mode Idle) – Point Target Lamp, State Transition

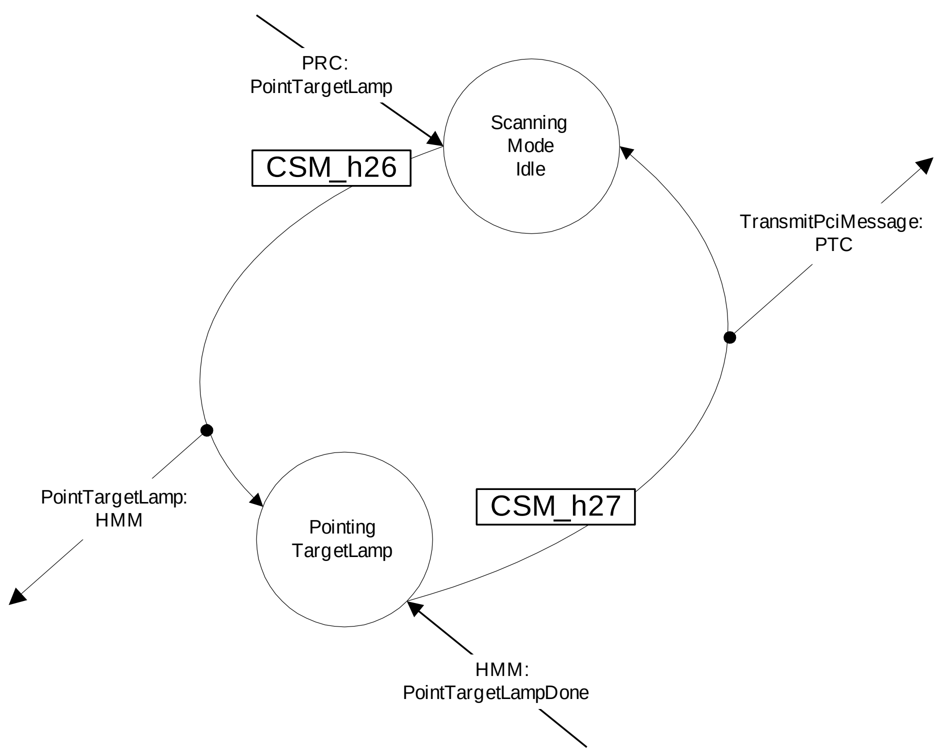
This message will position the center of the Target Lamp to the requested position. When in the Scanning Mode, the NT Application uses this to point to the "Alignment Point" that was successfully located.

This message is accompanied by data:

Data 1 = X Coordinate

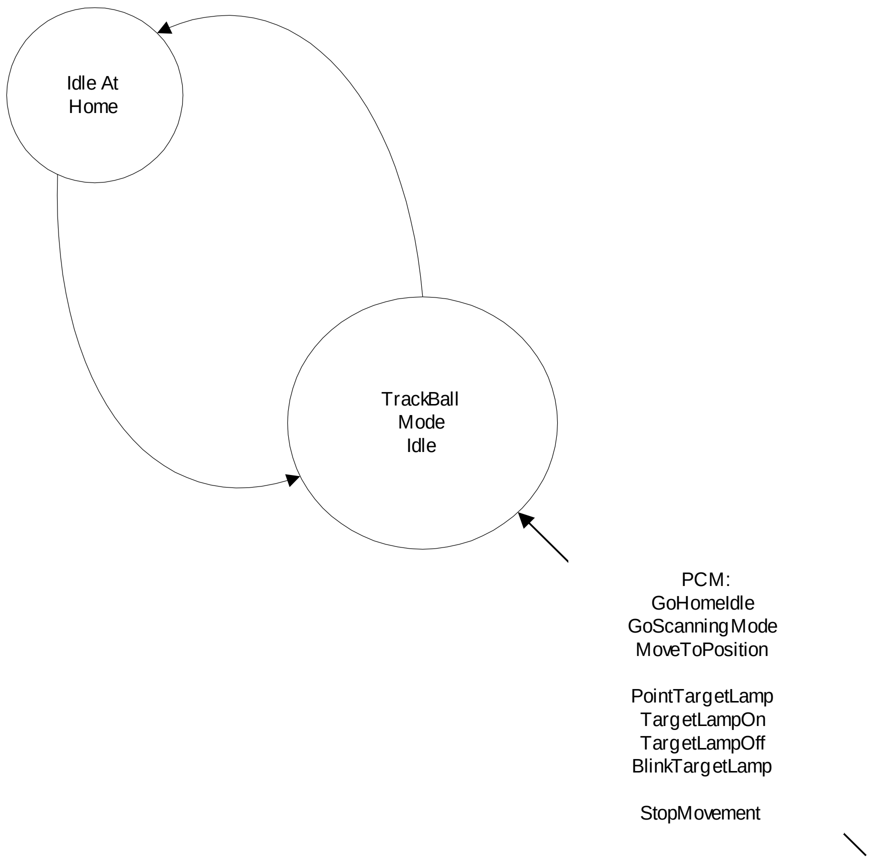
Data 2 = Y Coordinate

The state transition is simple. When a Point Target Lamp message is received, CSM simply tells HMM to perform the operation. The NT Application is notified when the process is complete. A "Flash Target Lamp" request is typically made by the NT Application after this one.



### 6.4.1.1.21. CSM – Track Ball Mode Idle, Messages Handled

The diagram below shows all the messages that CSM will handle while it is in the Track Ball Mode Idle State. The next pages will show how CSM handles each of these messages as well as the state transitions that occur based on the event.

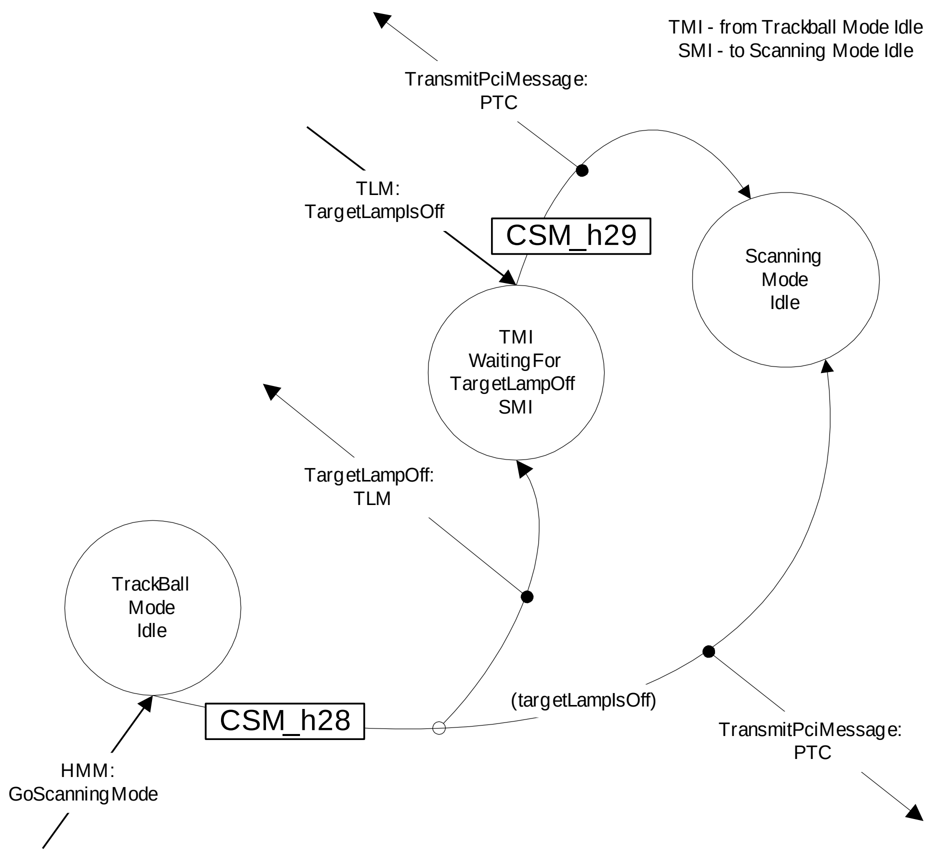



**IMPORTANT: The use of the Target Lamp and the Trackball comes hand in hand. All of the messages listed above will affect the state of the Target Lamp as seen throughout this section. Changing the mode from Trackball Mode to another will automatically turn off the Target Lamp, for instance.**

### 6.4.1.1.22. CSM – (Track Ball Mode Idle), Go Scanning Mode, State Transitions

This message allows changing the mode from Track Ball Mode to Scanning Mode. There are instances where sheet scanning needs to be executed right after some trackball operations. All “Scanning messages” or operations can then be executed once the mode change has been made.

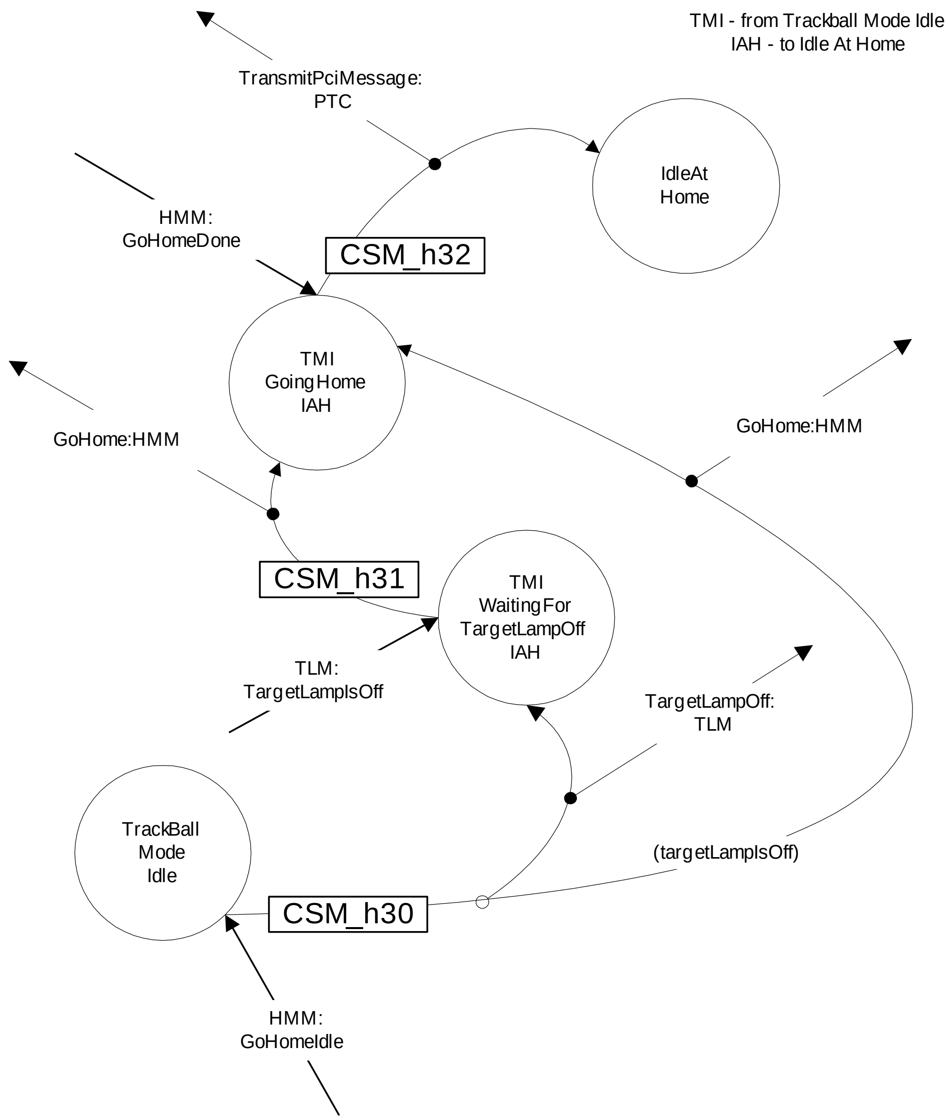
Note: CSM also keeps track of the Target Lamp State (On/Off) as requested by the NT Application. If the Target Lamp is on, it is automatically turned off with this procedure. The Target Lamp behaves differently while in the Scanning Modes.



### 6.4.1.1.23. CSM – (Track Ball Mode Idle), Go Home Idle, State Transitions

This message allows changing the mode from Track Ball Mode to A Home Idle Mode. This message normally ends a trackball session.

Note: If the Target Lamp is on, it is automatically turned off with this procedure. Target Lamp is off while in the At Home Idle State.



### 6.4.1.1.24. CSM (Trackball Mode Idle) – MoveToPosition, State Transitions

The NT Application sends the "Move To Position" message to perform the Trackball moves. The NT Application will also take into consideration the acceleration of the trackball as used by the operator, in order to make the probe head move as smooth as possible. This acceleration "profile" of the trackball itself is translated by means of sending different move distances - long moves if the trackball acceleration was high, and short moves if the trackball was only nudged.

**IMPORTANT:**

**As stated earlier, these messages affect the Target Lamp's state. The controller board will automatically turn off the target lamp after some time of trackball inactivity. It may automatically be turned on once activity resumes. This is true when any of these 2 messages are received.**

1. **MoveToPosition**
2. **PointTargetLamp**

**These messages may reactivate the Target Lamp themselves. A flag called "ntTargetLampOn" keeps track of the "Target Lamp On/Off" requests made by the NT Application while in the Trackball Mode.**

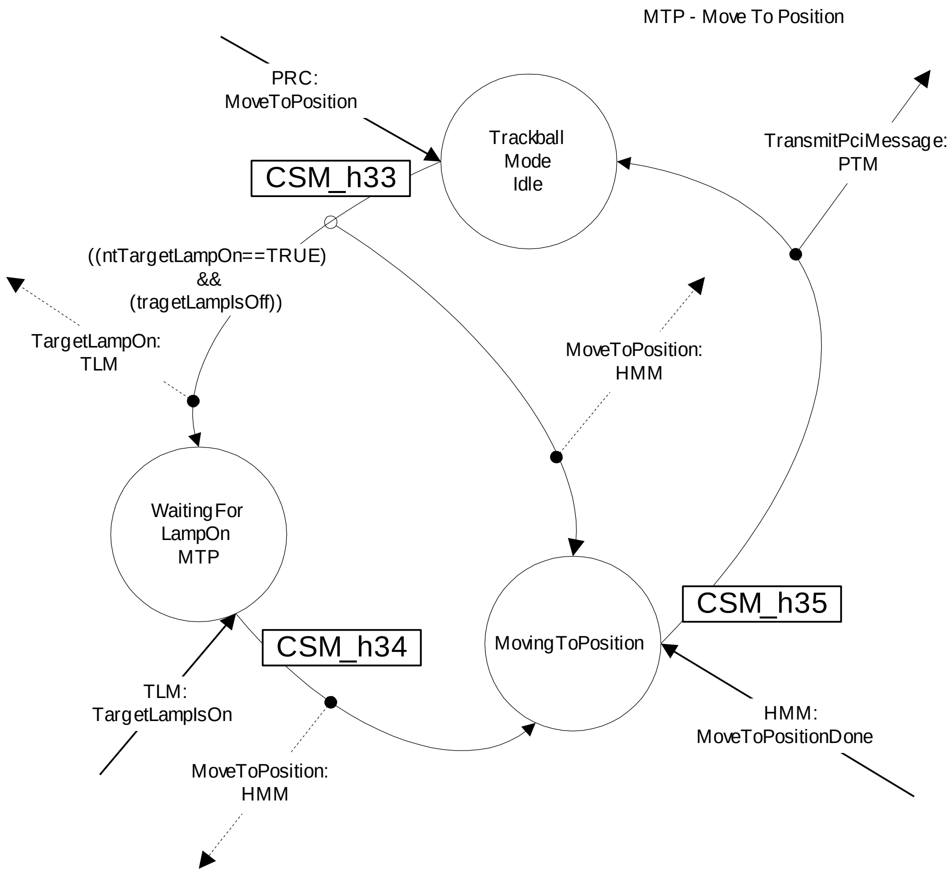
**This implies that the NT Application needs to make a "Target Lamp On" request if he wants the Target Lamp to be active. The flag ensures that "Target Lamp On" request is properly sent to TLM to reactivate the Target Lamp if it was automatically turned off by TLM.**

### 6.4.1.1.25. CSM (Trackball Mode Idle) – MoveToPosition, State Transitions continued . . .

PRC sends the MoveToPosition message with data.

Data 1 - X Measurement Sensor coordinate
Data 2 - Y Measurement Sensor coordinate

CSM then relays the message to HMM. Once the move is complete, HMM will send a message to CSM about this event. CSM then informs the NT Application that the move was completed.




Notes:

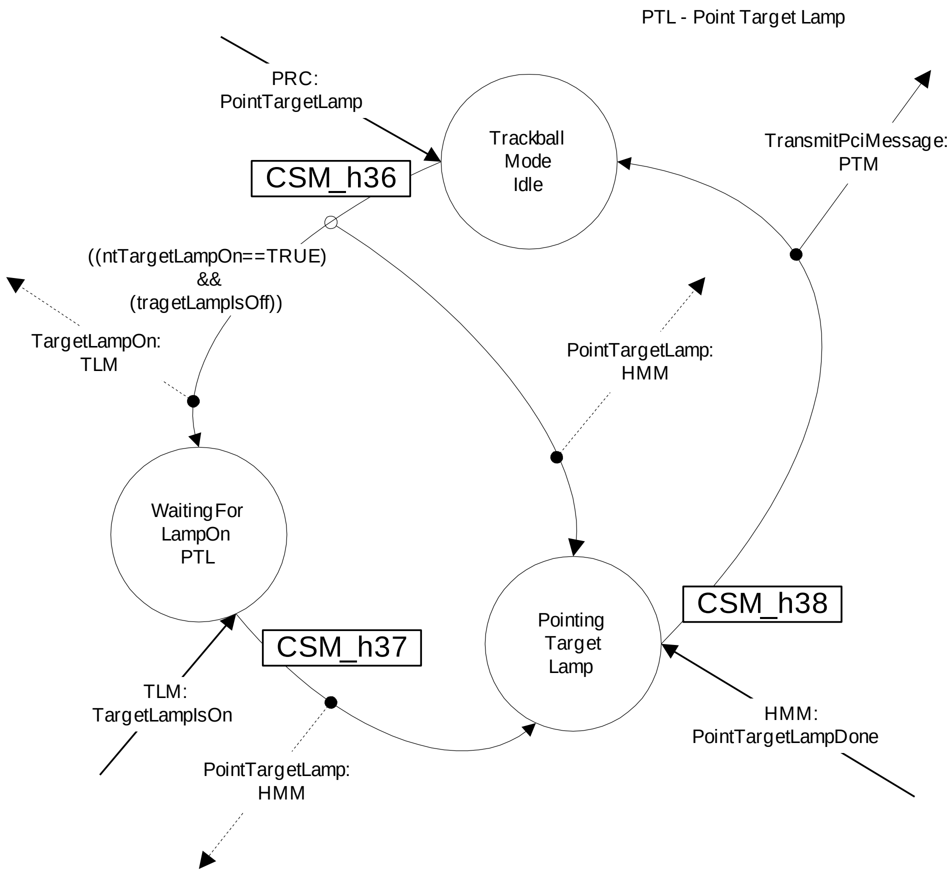
- Backlash compensation is disabled in this mode. The NT Application will keep track of the moves and the their backlash values. These values will be needed to determine the precise point location when the actual measurement takes place.
- Trackball moves uses only one move profile

### 6.4.1.1.26. CSM (Trackball Mode Idle) – Point Target Lamp, State Transition

This request is similar to the “Move To Position” message except that the position requested will be the center of the Target Lamp instead of the measurement sensor.

PRC sends the Point Target Lamp message with data.

Data 1 - X Target Lamp coordinate
Data 2 - Y Target Lamp coordinate

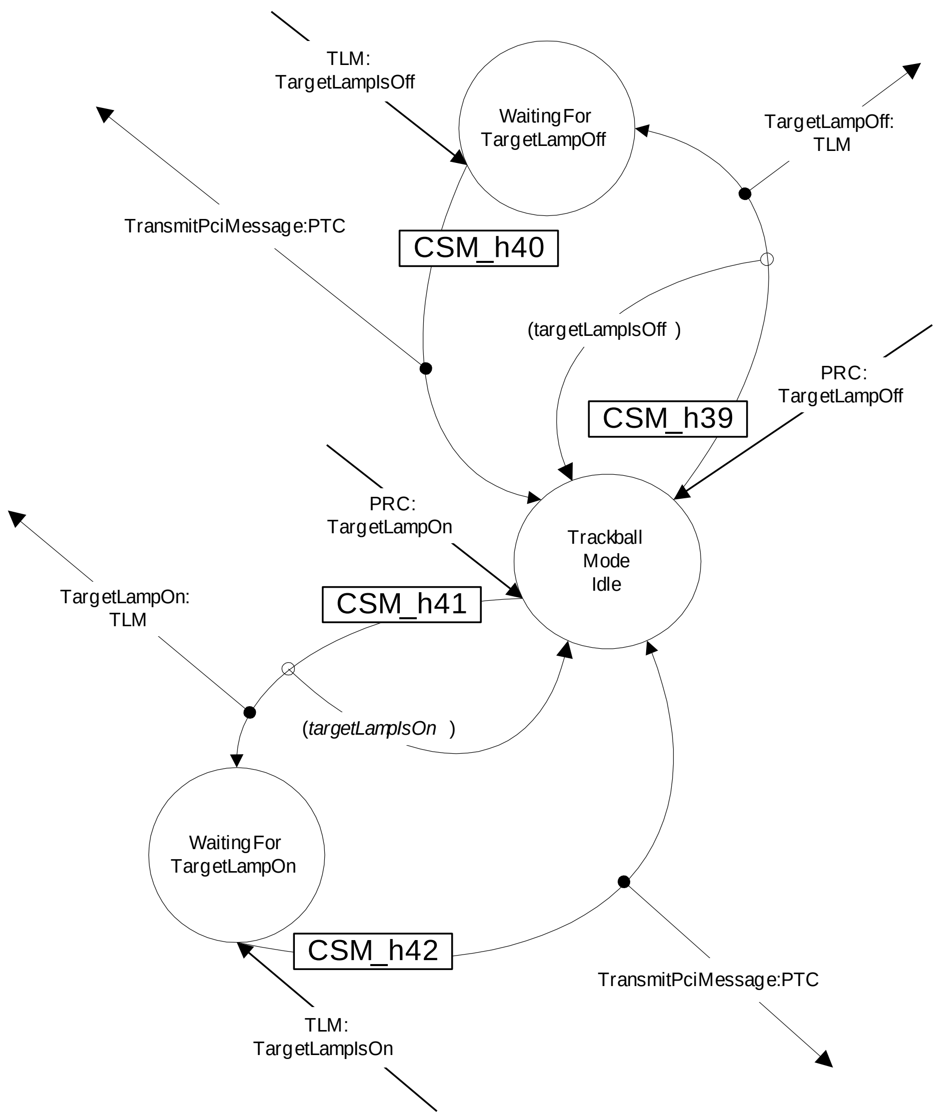



Notes:

- Backlash compensation is disabled in this mode. The NT Application will keep track of the moves and the their backlash values. These values will be needed to determine the precise point location when the actual measurement takes place.
- Trackball moves uses only one move profile

### 6.4.1.1.27. CSM – (Trackball Mode Idle), Target Lamp On, Target Lamp Off, State Transitions

The diagram below shows the transition when either of the 2 Target Lamp messages are received. When these requests are received, CSM simply forwards it to TLM if necessary, and waits until the request has been completed by TLM. The NT Application is informed about the completion of the request.

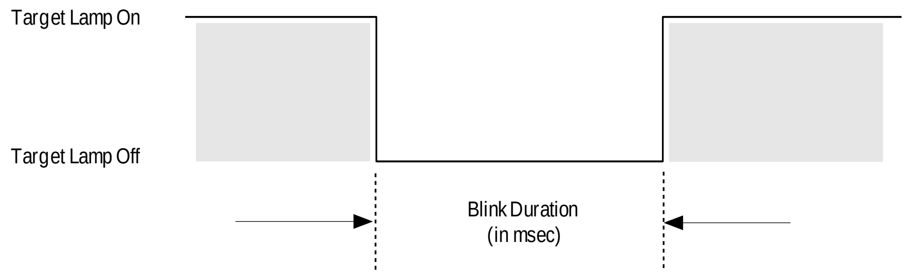


### 6.4.1.1.28. CSM – System Interaction, Trackball Mode, Blink Target Lamp

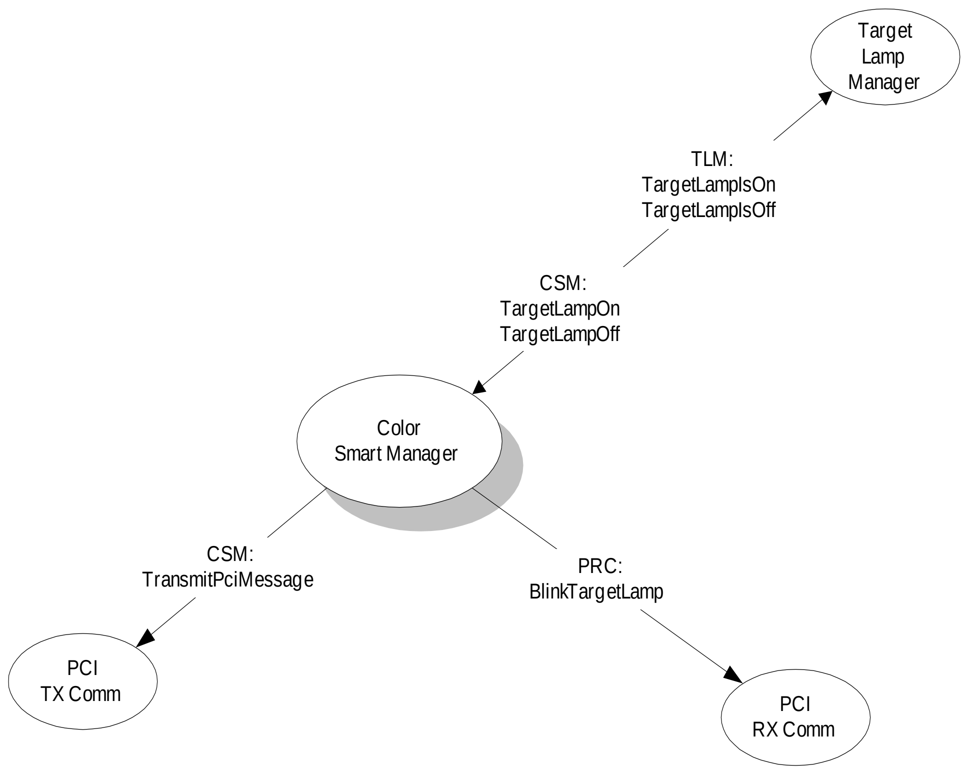
The Blink Target Lamp request is normally used by the NT Application to acknowledge that the end user has “clicked” the trackball button.

Note: The NT Application will have to wait until the “Blink Target Lamp” process is complete before it sends the next operational request – when CSM is back in the Trackball Mode Idle State.

Blink Target Lamp Timing Diagram:




Below is a diagram that shows the machines that are involved in this process.



**IMPORTANT: In order to see the "Blink" effect, a "Target Lamp On" request must be made prior to the Blink Target Lamp request.**

**This request will leave the Target Lamp activated by the time the process is completed. This message is treated as if a Target Lamp On message was sent.**

**CSM uses the max Target Lamp On Time when it sends the final "Target Lamp On" request as seen in the state transition diagram.**

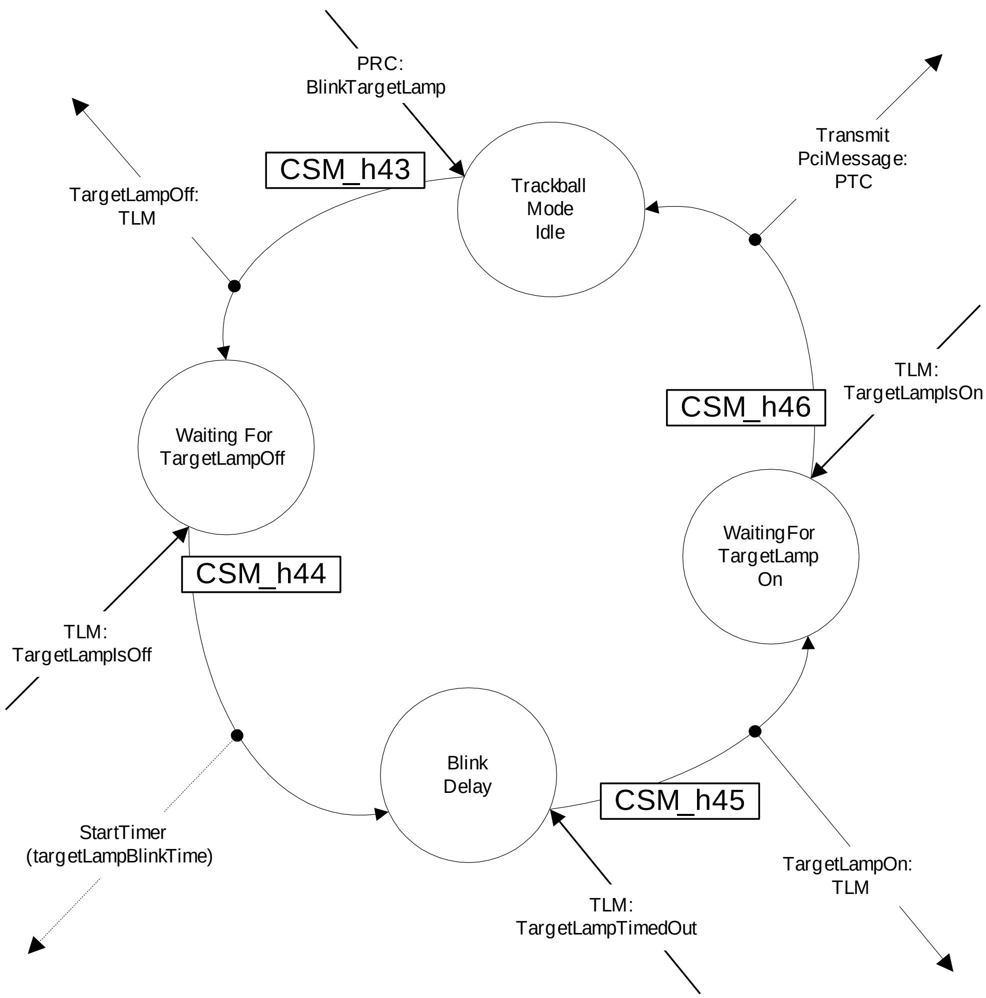
### 6.4.1.1.29. CSM – (Trackball Mode Idle), Blink Target Lamp, State Transitions

When a "Blink Target Lamp" request is received, CSM in turn makes a request to TLM to turn off the lamp for a given duration - *"Blink".* A Timer is started for the duration that the Target Lamp should be off. A Target Lamp Off request is then made when the TimeOut event is received. The NT Application is finally informed that the request is complete when TLM says the Target Lamp is on.

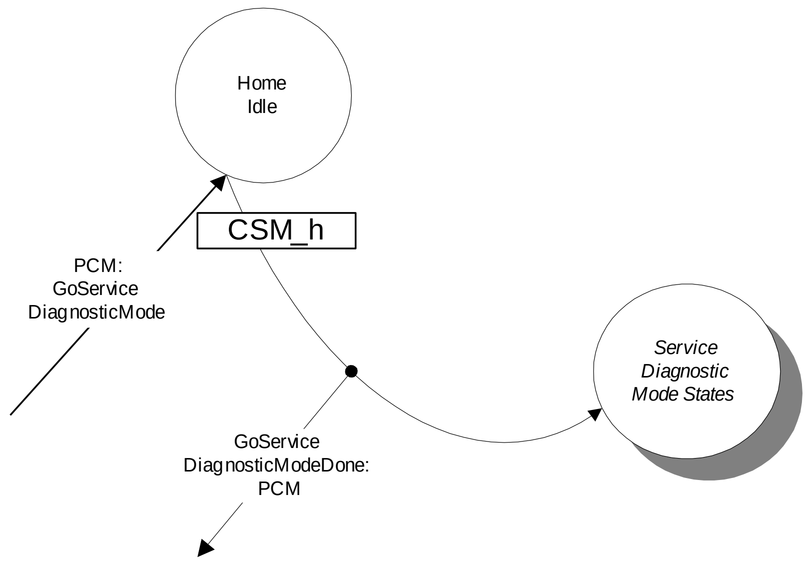
The Blink Target Lamp message is accompanied by data:

Data 1 = Target Lamp Blink (Off) Time - contain how long the target lamp stays off (in 100 msec units).

Data 2 = Unused

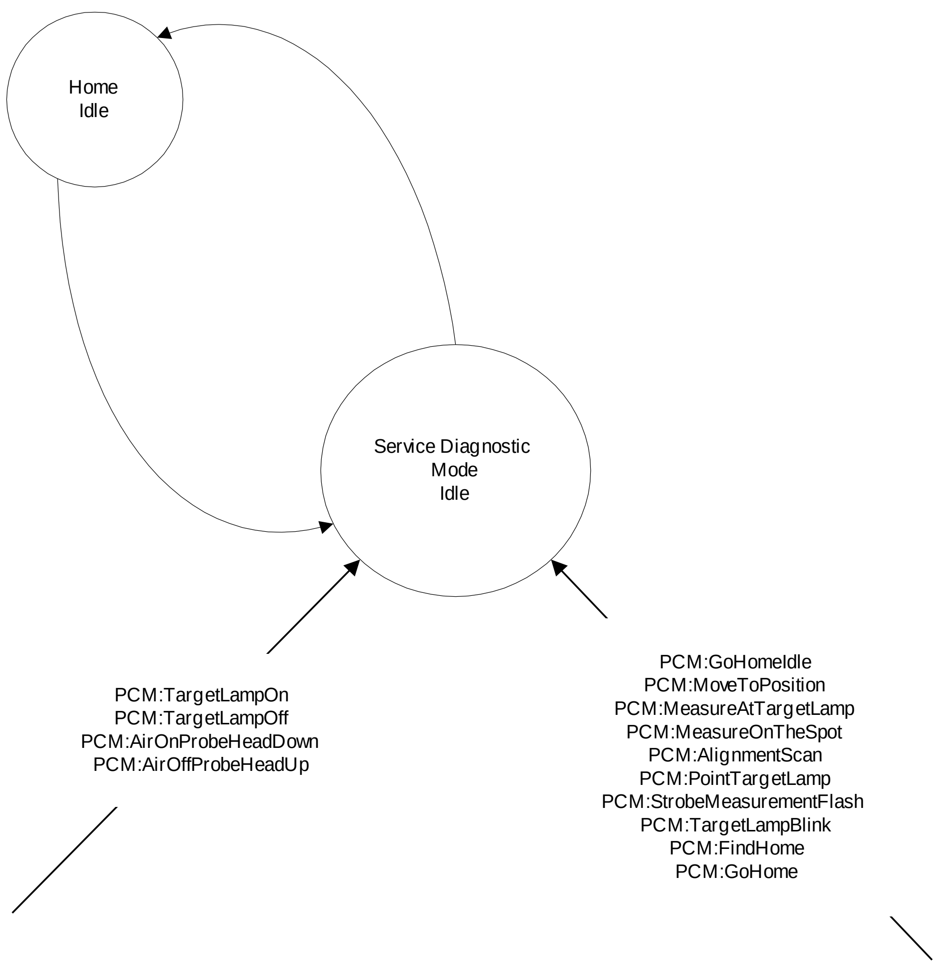


## 6.4.1.1.30. CSM – Go Development Diagnostic Mode, State Transitions

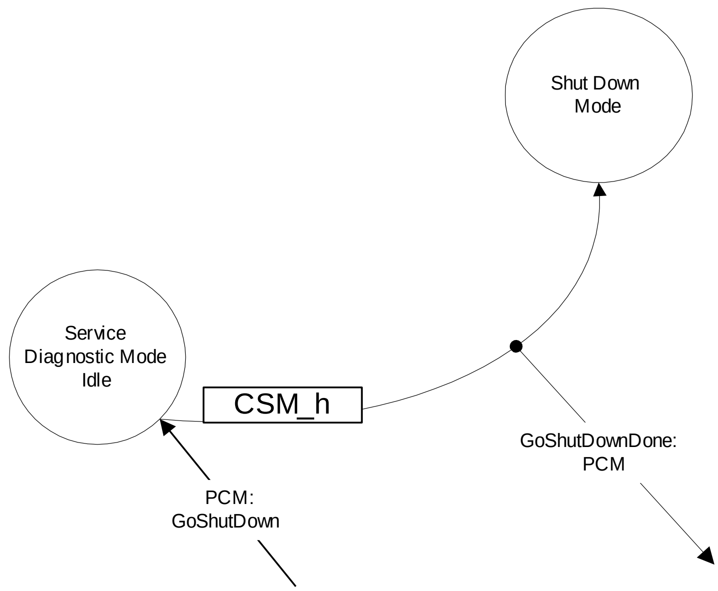


### 6.4.1.1.31. Development Diagnostic Mode Idle, Messages Handled

The diagram below shows all the messages that CSM will handle while it is in the Service Diagnostic Mode Idle State. The next pages will show how CSM handles each of these messages as well as the state transitions that occur based on the event.

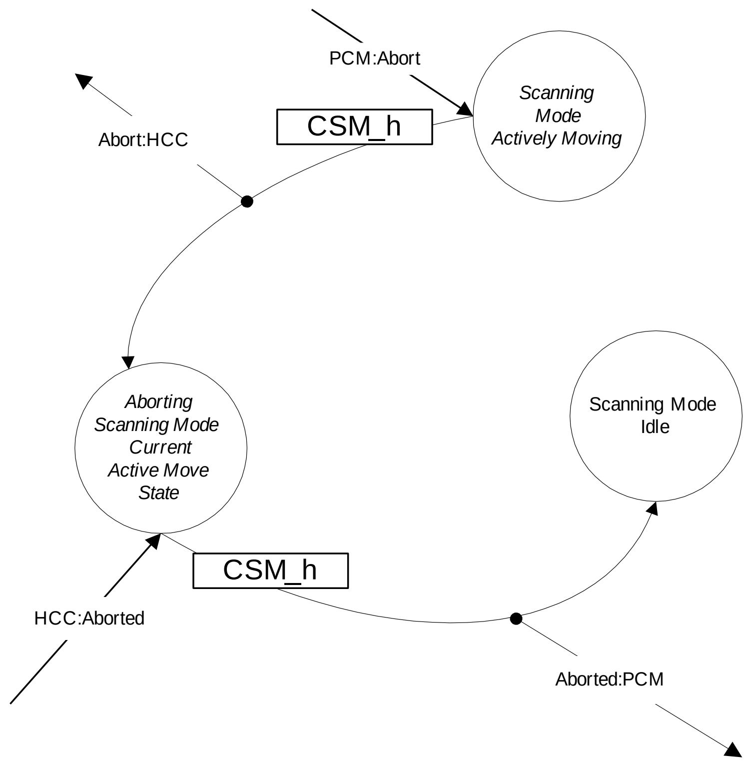


### 6.4.1.1.32. CSM (Development Diagnostic Mode Idle) – Go Shut Down, State Transitions



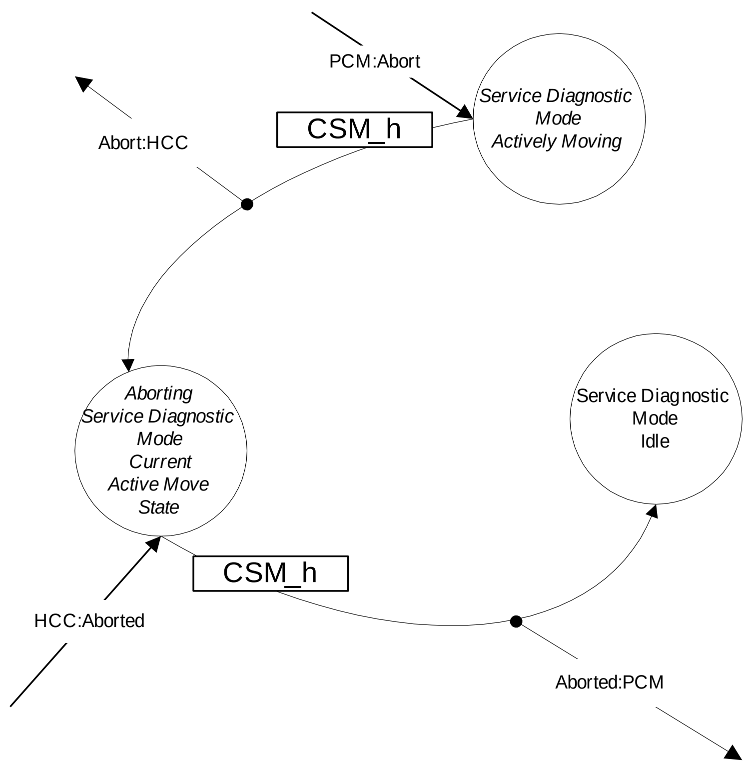
### 6.4.1.1.33. CSM – (Scanning Mode Actively Moving), Abort, State Transitions

The “Abort” message is handled at any state where the probe head might be actively moving while in the Scanning Mode States. This message forces CSM to go back into Scanning Mode Idle State.

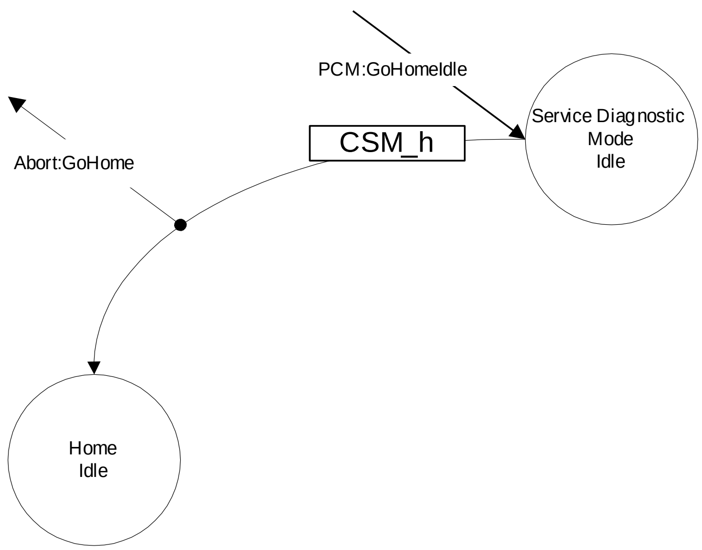


### 6.4.1.1.34. CSM – Service Diagnostic Mode, Abort, State Transitions

The “Abort” message is handled at any state where the probe head might be actively moving while in the Service Diagnostic Mode States. This message forces CSM to go back into Service Diagnostic Mode Idle State.

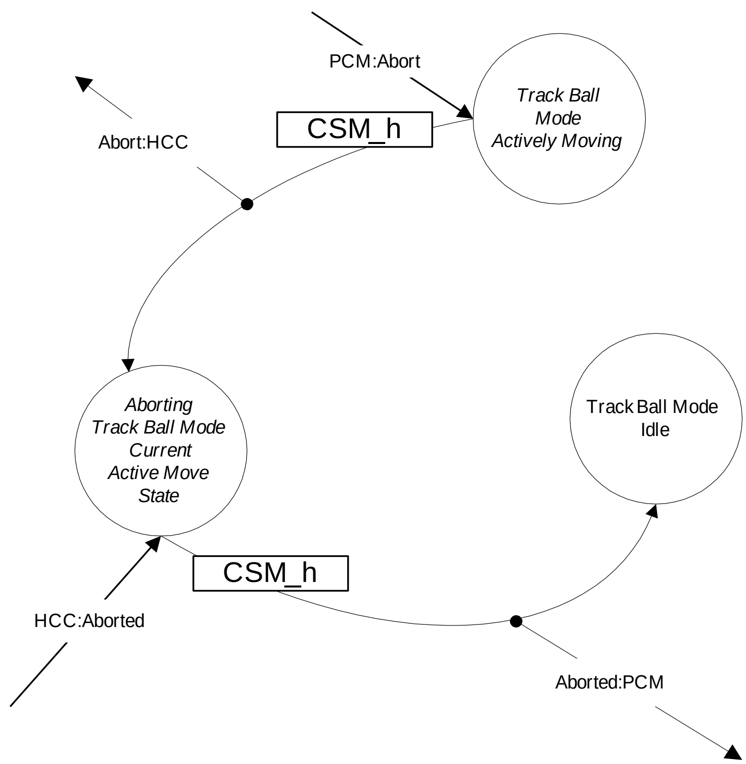


6.4.1.1.35. CSM – Service Diagnostic Mode, GoHomeIdle, State Transitions



### 6.4.1.1.36. CSM – Track Ball Mode, Abort, State Transitions

The “Abort” message is handled at any state where the probe head might be actively moving while in the Track Ball Mode States. This message forces CSM to go back into Track Ball Mode Idle State.



### *6.5. Synchronous Comm IRQ (ISCM) Task Description*

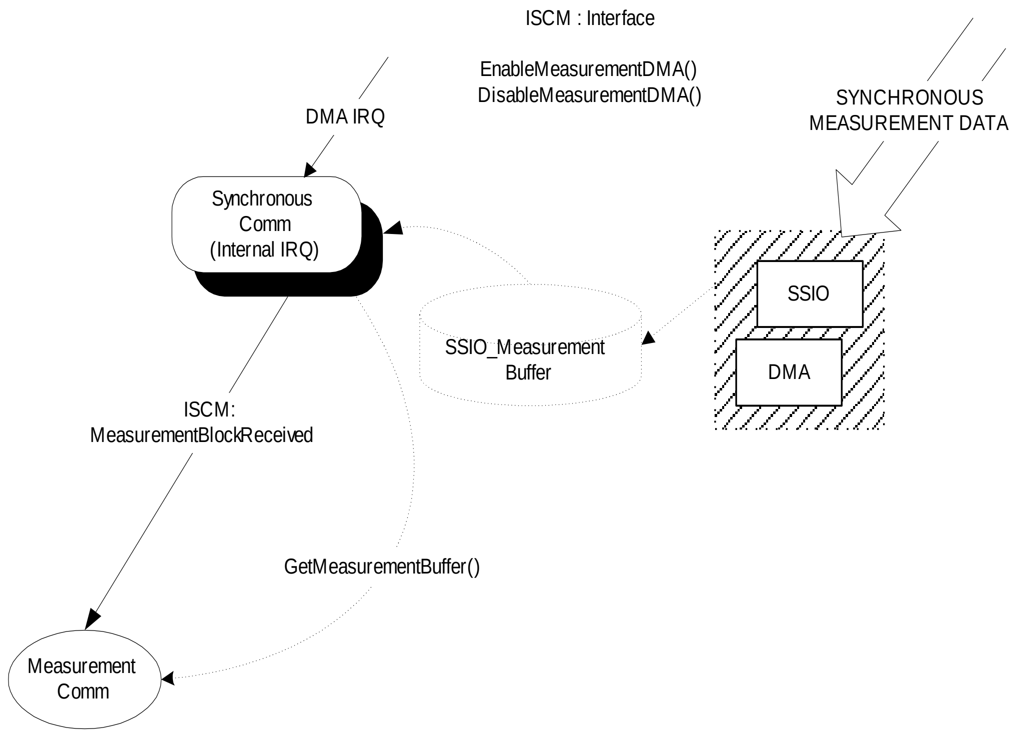
***ISCM - I****nterrupt* ***S****ynchronous* ***C****om****M***

ISCM is an interrupt service routine that will handle all the low level processing of the Synchronous (Data + Clock @1 MBaud) Interrupts that will be generated by the incoming scan measurement data.

ISCM utilizes the DMA feature of the processor to fill the SSIO (Sync Serial I O) data into a buffer. Once the buffer is full, the DMA issues an interrupt (indicating Buffer is Full). Through the Kernel, MCM receives the message notifying him about the event. MCM then takes the Measurement Data to perform further processing (if needed).

Note: ISCM owns the measurement buffer. As seen in the diagram below, MCM uses an function to access this measurement buffer.

#### 6.5.1. ISCM : MCM - Data Flow Diagram




Note:

The use of the DMA feature allows us to handle these High Baud Rates data transfers. Instead of getting an interrupt every WORD reception, we only get one at the end of the 512-Byte reception.

### *6.6. Measurement Comm (MCM) Task Description*

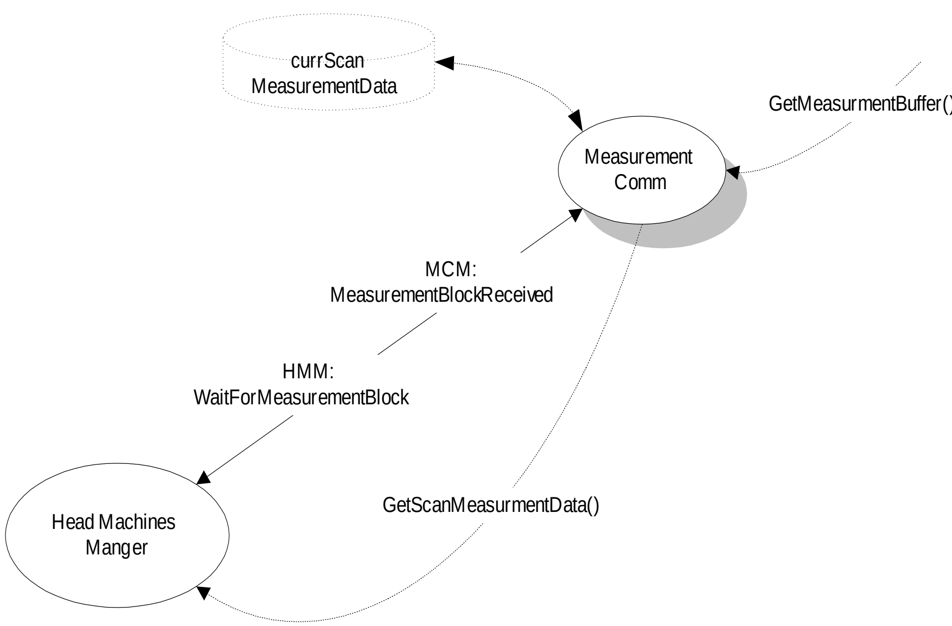
#### ***MCM - M**easurement **C**om**M***

This will handle all the measurement data transferred between the Head Scanner and the Color Smart PCI Card. The state machine will handle the task based on a Communication Protocol. The task will include Timeout detection where the timing services of the OS Kernel will be utilized.

The MCM will provide buffering of 1 Measurement Data (512 Bytes). The arrival (event) of the measurement data is relayed to the Head Machines Manager (HMM). HMM in turn gets the data from MCM to perform further processing.

**IMPORTANT: HMM is in charge of retrieving the data from MCM. HMM uses the services of the Measurement Data Manger to store the individual measurement data into the Shared Memory. There is enough time in the Color Smart Design to allow data transfer between the associated the machines.**

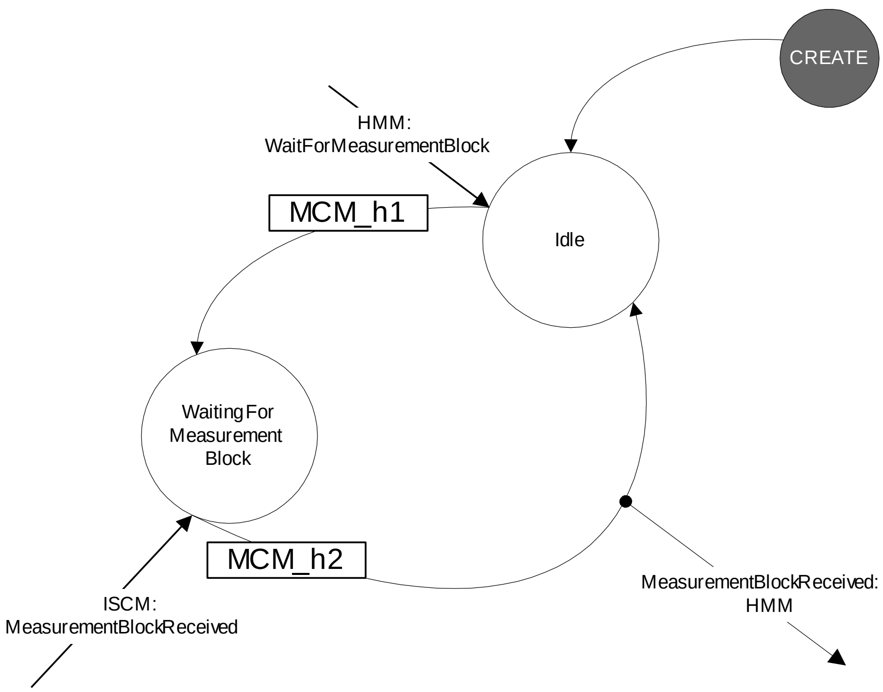
#### 6.6.1.1. MCM – System Interaction, WaitForMeasurementBlock




Note: MCM has his own copy of the measurement data.

MCM can also perform validation of the Measurement Data, which may include Flash Error Detection, Timeout Detection or Checksum Detection.

6.6.1.1.1. MCM – WaitForMeasurementBlock, State Transitions



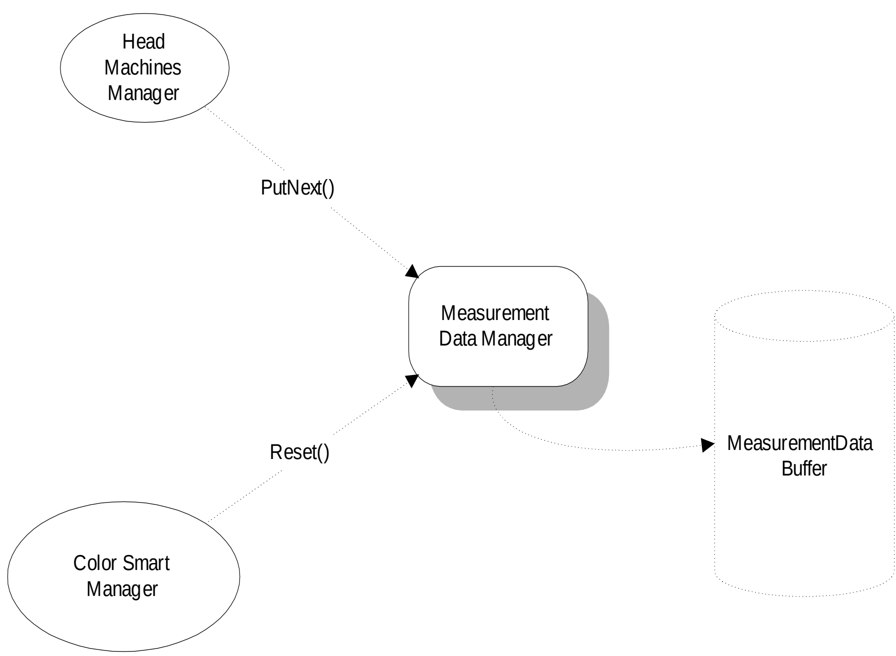
### *6.7. Measurement Data Manager (MDM) Task Description*

***MDM*** *-* ***M****easurement* ***D****ata* ***M****anager*

Similar to the previous manager discussed, this manager keeps a database of the measurement data. The Head Coordinator provides data to be stored. This measurement data is primarily used by the NT Application, which has direct access to it through the shared memory.

**IMPORTANT: CSM is the only machine who should reset MCM. Resetting MCM is generally done just before any measurement session is initiated.**

#### 6.7.1. MDM – High Data Flow Diagram



### *6.8. Asynchronous Comm IRQ (IACM) Task Description*

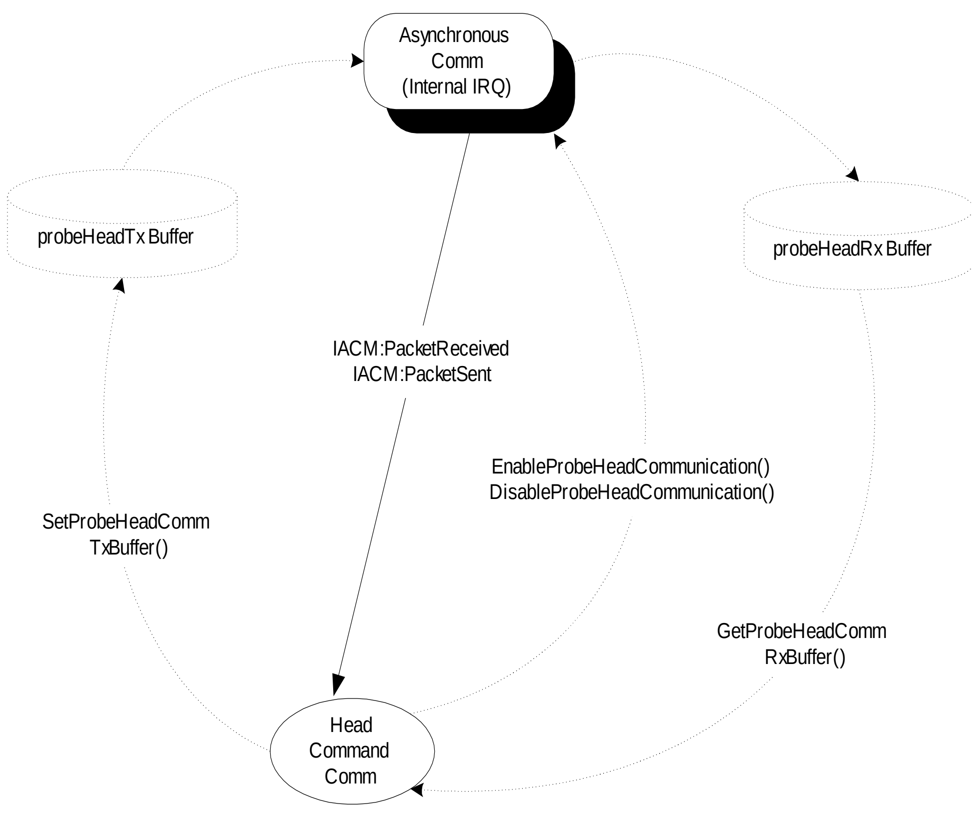
### ***IACM - I**nterrupt **A**synchronous **C**om**M***

This is an interrupt service routine that will handle all the low level processing of the Asynchronous Interrupts that will be generated by the incoming replies of the Probe Head. The communication protocol uses RS232 running at 62.5Kbaud.

When enabled, this ISR sends the contents of the Tx Buffer. It uses the Rx Buffer to receive the incoming data. IACM locally owns these two communication buffers shown in the diagram below. HMM uses access functions to set or retrieve these buffers.

Note: The Receive and the Transmit interrupts are only enabled when they need to be used. They are disabled when they are not needed. HCC is responsible for enabling or disabling IACM.

#### 6.8.1. IACM : HCC - Data Flow Diagram



## *6.9. Head Command Comm (HCC) Task Description*

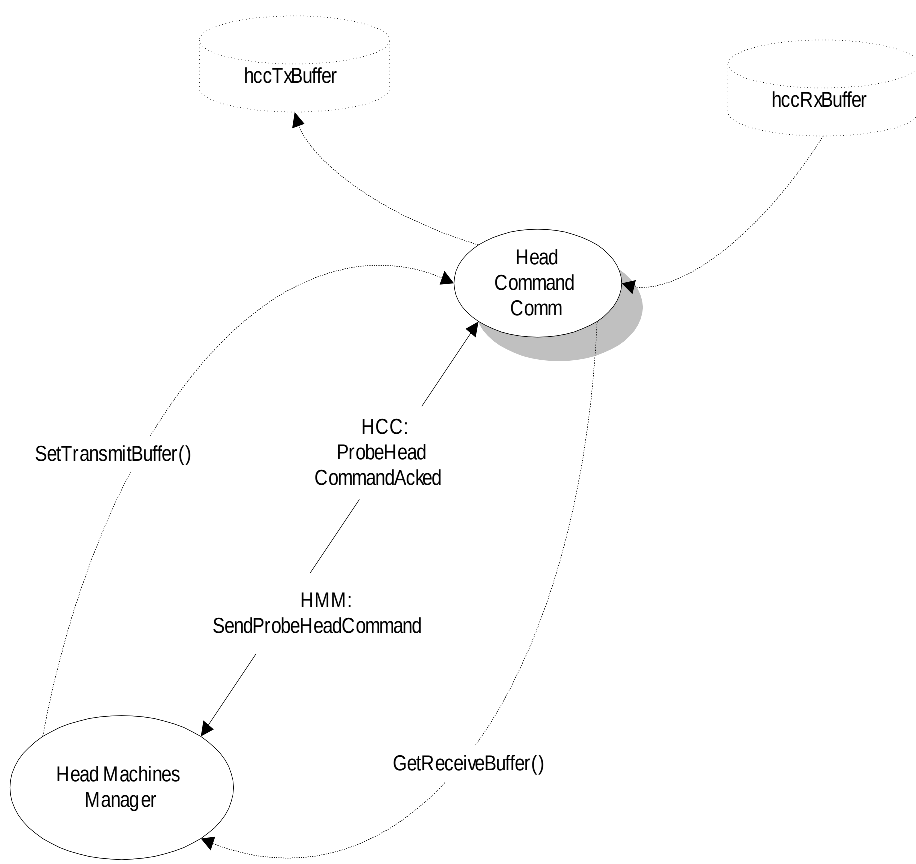
### ***H**CC - **H**ead **C**ommand **C**omm*

This SM will handle all the communication that is transacted between the Head Scanner and the Color Smart PCI Card. The state machine will handle the task based on a Communication Protocol which include checksum generation among other things. The task will include Timeout detection where the timing services of the OS Kernel will be utilized.

The Probe Head operates as a slave of the Color Smart Controller Card. It will only respond when requested. In this case a “half-duplex” form of communication is followed.

The diagram below shows that HMM uses access functions to set or retrieve the communication buffers of HCC.

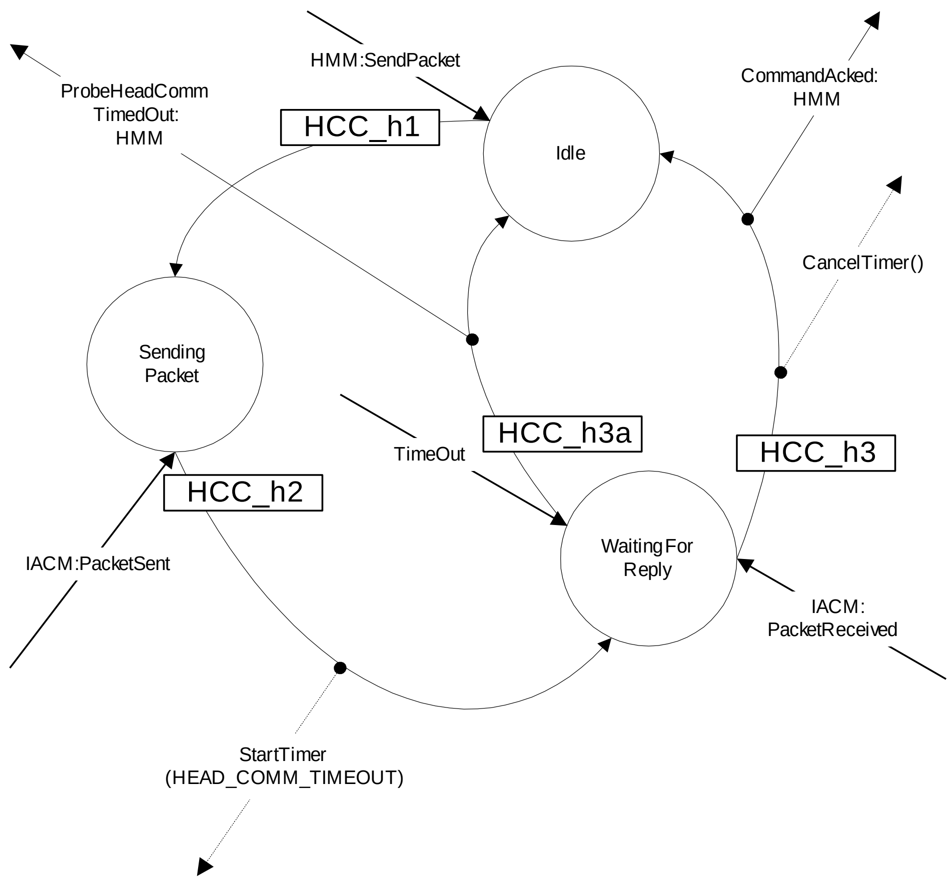
### 6.9.1. HCC – High Level Context Diagram



### 6.9.1.1. HCC – SendPacket, State Transitions

The SetTransmitBuffer access function is used to fill up the TXBuffer. A request is then made to HCC to initiate the transfer.

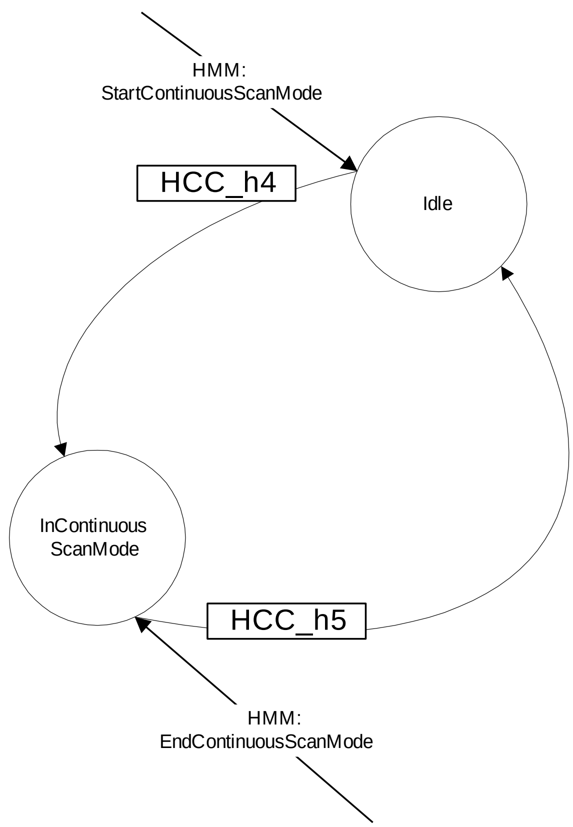
The head communication protocol specifies that each command to the head will generate a reply. This is why the state transitions come from the Sending State and goes directly to Waiting For a reply State.



### 6.9.1.2. HCC – StartContinuousScanMode, EndContinuousScanMode - State Transitions

The color bar scan requires precise timing. Due to this the HCC has been designed to have a state where it does not accept any message but an EndContinuousScanMode message.

To limit latencies induced by the Kernel, the “Measure Command” will be sent at the interrupt level. See Motor Chip Set Comm (IMCSC) on page 72 for more detail.



### *6.10. Motor Chip Set Comm (IMCSC) Task Description*

#### ***IMCSC - I****nterrupt* ***M****otor* ***C****hip* ***S****et* ***C****omm*

This is an interrupt service routine that will handle all the low level processing of an external Interrupt that will be generated by the completion of a motor move or any other interrupts generated by the Motor Chip Set.

**IMPORTANT:**

**The Color Smart System has only one "Hard Real Time" requirement.** This is required when CS is in the process of scanning a Color Bar where the Probe Head moves at a constant speed across the bar.

When the system is in the Color Bar Scan Mode, IMCSC plays a special role to meet the timing demands of the Color Bar Scan process. In this mode IMCSC bypasses the Kernel Scheduling Services and directly instructs the Asynchronous ISR (IACM) to send the measurement command to the Probe Head. The measurement commands are sent at specific Probe Head position intervals. To provide the highest positional accuracy, the Motor Chipset will be programmed to generate an interrupt at these intervals. These are called "break points" in Motor Chip Set terms. When a break point interrupt is generated, IMCSC immediately activates the IACM to send the measurement command to the Probe Head. All of these are accomplished at the interrupt level to minimize latency and maintain accuracy.

Also see HMM – System Interaction, DoColorBarScan on page 149 for more information about the Color Bar Scan process.

Note: HMM knows when a color bar scan is being performed. Therefore, all machines within the Head State Machine Complex will still be synchronized while performing this process.

### *6.11. Motor Comm (MTRC) Task Description*

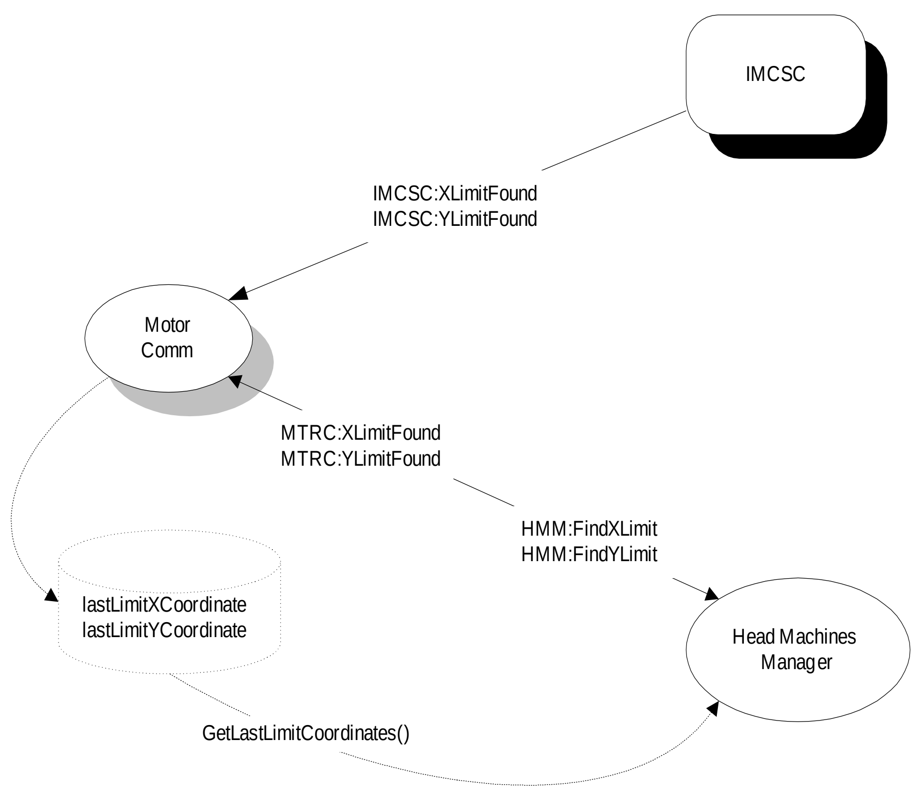
### ***MTRC - MoToR Comm***

This SM will handle all the communication that is transacted between the Motor Chip Set and the 386EX I/O Data Port. The state machine will handle the task based on a Communication Protocol provided by the Motor Chip Set manufacturer. The task will include Timeout detection where the timing services of the OS Kernel will be utilized.

MTRC receives messages directly from HMM. In order for the motors to change it position, HMM would set the target coordinates and then send a GotoXYTarget Message. HMM will receive an "XYAxisOnTarget" event when both axis are in the requested position.

Finding the limits, as a positional reference is another service provided by MTRC. This allows the Color Smart to position the probe head accurately by using the limit references. When looking for the "limits" position, the motor chip sets are programmed to immediately stop the motor as soon as the limit switch is hit. At this point the motor chipset also generates an interrupt processed by IMCSC. IMCSC in turn sends a message to MTRC about the "hitting the limit" event.

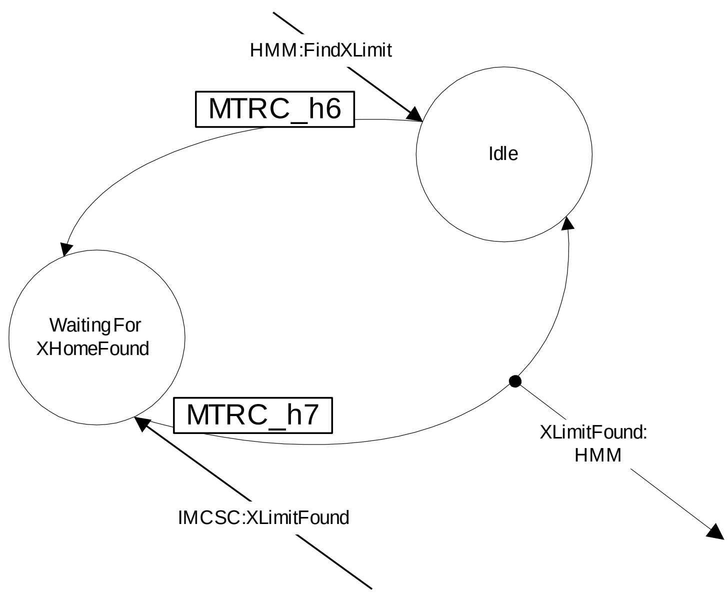
#### 6.11.1.1. MTRC – System Interaction, FindLimits



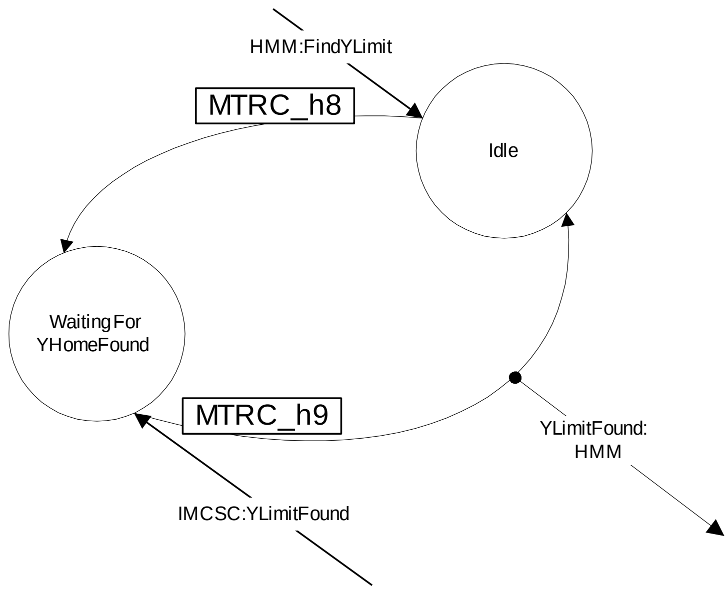
#### 6.11.1.1.1. MTRC – FindXLimit, State Transitions

Once MTRC receives the "Find Limit" message request, it makes all the necessary motor commands to force the probe head to hit the limit and stop. The machine stays in the "Waiting" state while the probe head is moving towards the limit. A "Limit Found" message takes the machine back to idle. HMM is notified about the completion of the request.

**IMPORTANT: The Find Limit process updates a variable called "lastLimitCoordinate". This is used by the NT Application to determine how much the limit reference position changes from time to time. This value can also be used to force a re-calibration if necessary.**



### 6.11.1.1.2. MTRC – FindYLimit, State Transitions



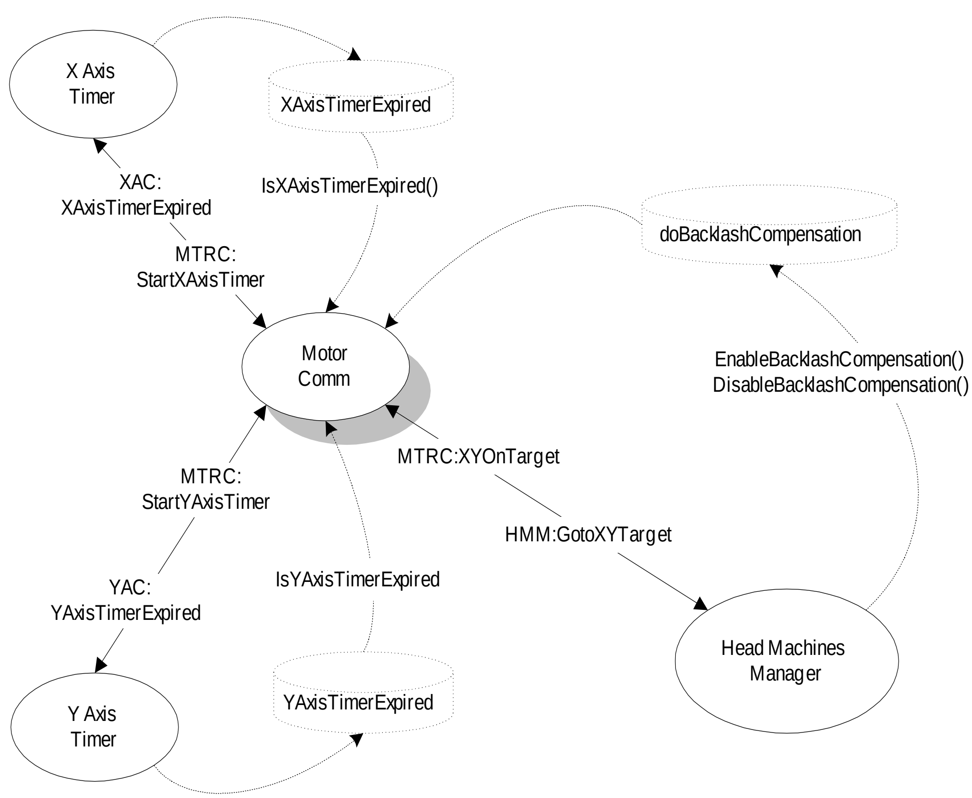
## 6.11.1.2. MTRC – System Interaction, GotoXYTarget

The "GotoXYTarget" is a message sent by HMM. This message is accompanied with data:

Data1 = target X Coordinate
Data2 = target Y Coordinate

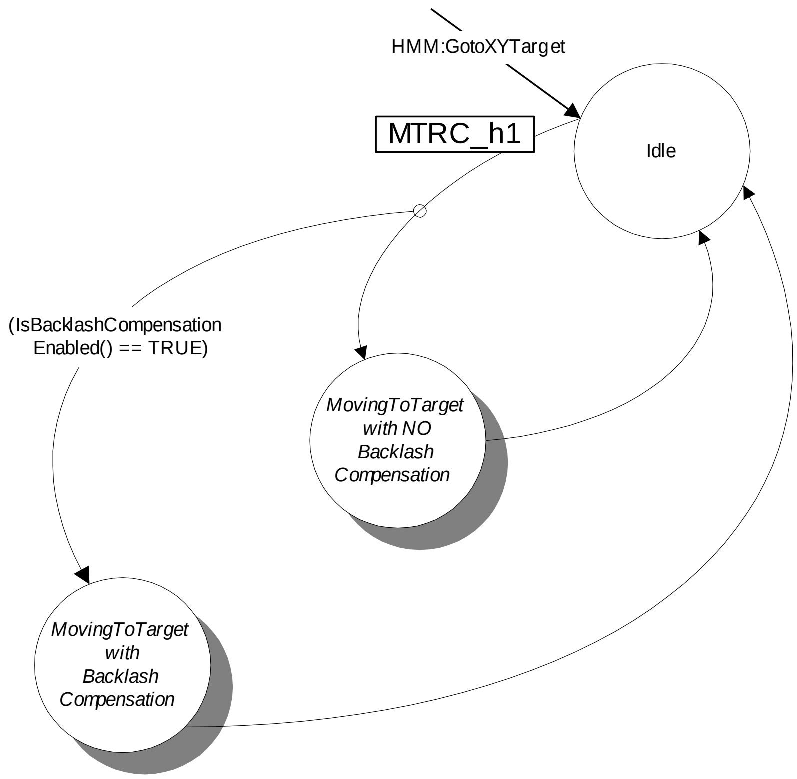
The backlash compensation is explicitly enabled or disabled by HMM just before sending the GotoXYTarget request.

**IMPORTANT: It is up to HMM to provide the "final settle" delay. This is the delay needed to make sure that the head is no longer oscillating as it settles into a the final position.**



### 6.11.1.2.1. GotoXYTarget, State Transitions – Level 1

Depending on the setting of the backlash compensation flag, MTRC can take one of 2 general states as shown below. These general states are further decomposed in the next pages.



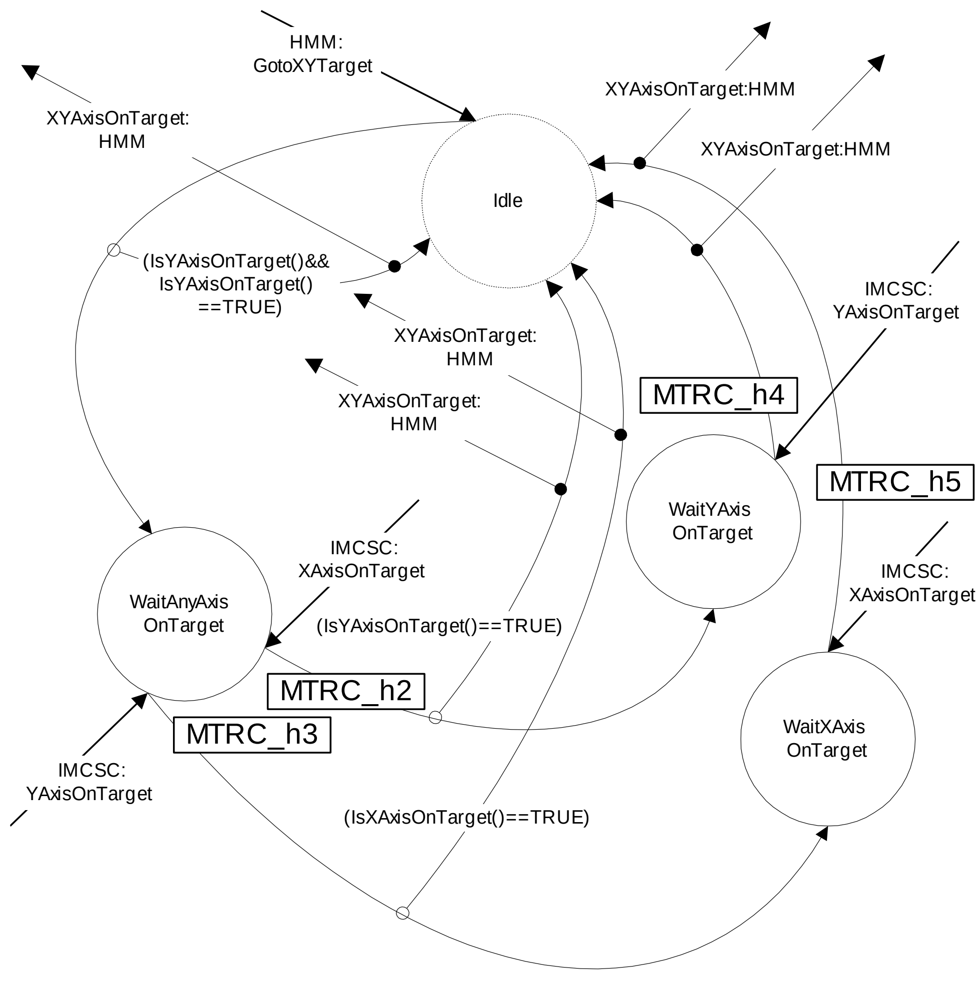
### 6.11.1.2.2. MTRC – GotoXYTarget, State Transitions (Backlash Compensation Disabled) – Level 2

The diagram below shows the different state transitions MTRC takes when backlash compensation is disabled.

NOTE:

MTRC enters the "WaitAnyAxisOnTarget" only when a move needs to be made. Before the machine enters the "WaitAnyAxisOnTarget" state, a move command is sent for the appropriate axis.

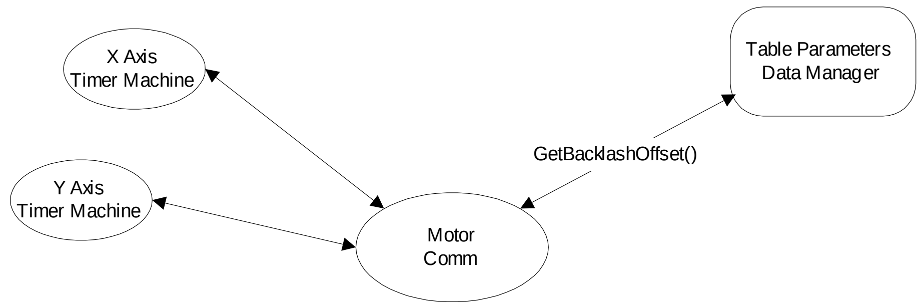
The reason for having 2 wait states, 1 for X and 1 for Y is that one axis can finish ahead of the other. The individual states accurately reflects what the SM is doing.



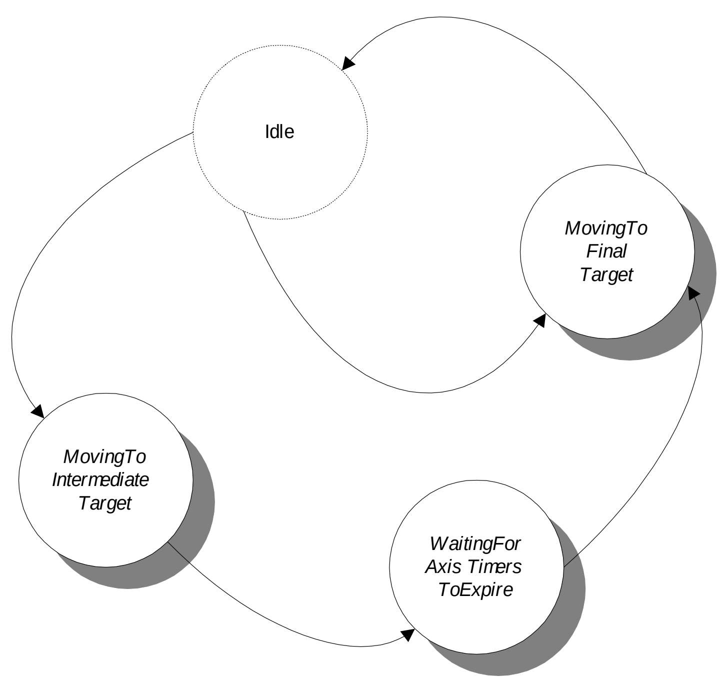
### 6.11.1.2.3. MTRC – GotoXYTarget, State Transitions (Backlash Compensation Enabled) – Level 2

When moving to the final target, the stepper motors need a special delay, if it has to move in the reverse direction. This delay is required to prevent the stepper motors from slipping due to the direction reversal.

The diagram below shows that we use a timer service when the move with Backlash Compensation Enabled is executed. The machine may receive a timeout event while it is in any of the Moving to Intermediate Target states. This relationship is better described in the X and Y Axis Timer sections.



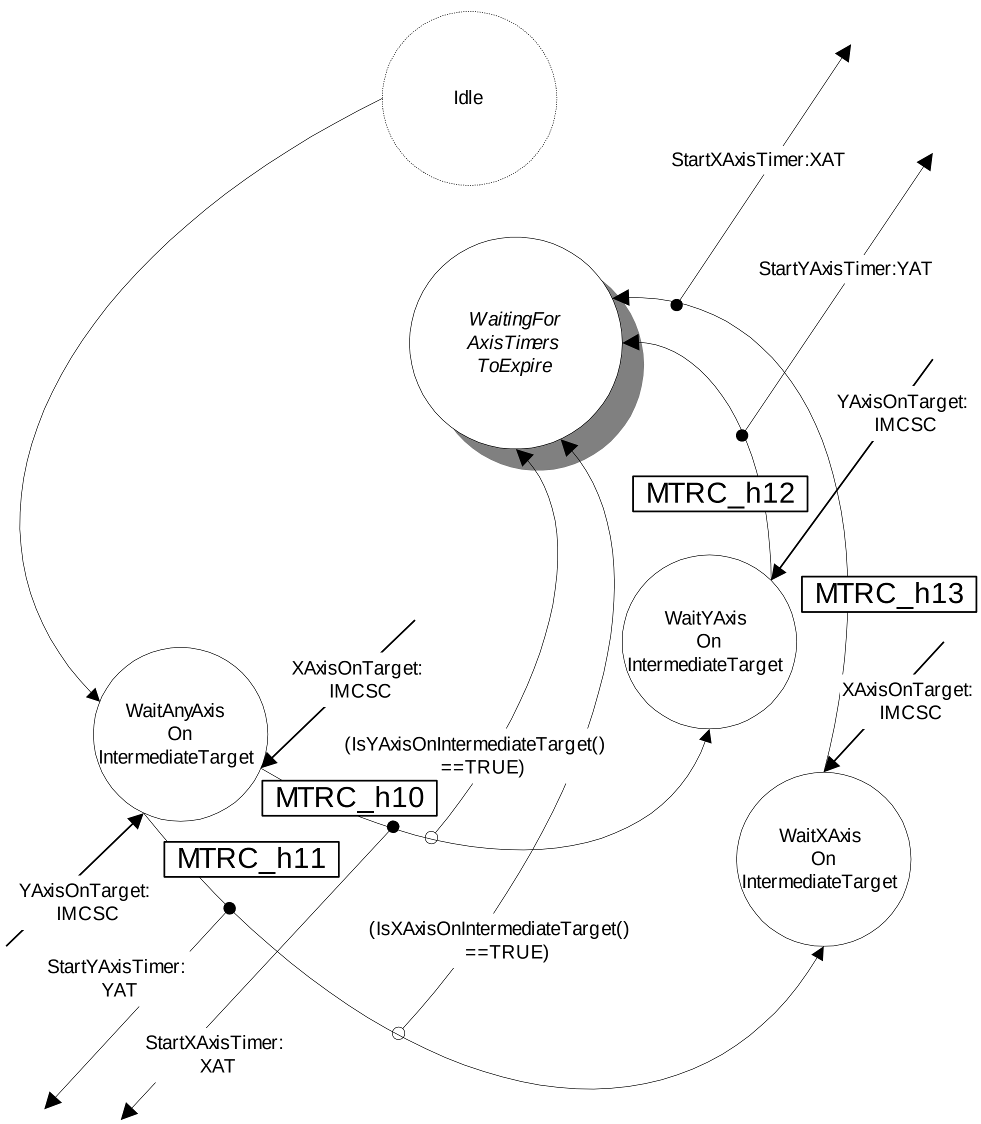

Due to the complexities of the backlash moves, the states are further divided into three groups. These groups are shown in detail in the next pages.



### 6.11.1.2.4. MTRC – GotoXYTarget, State Transitions (Backlash Compensation Enabled) – Level 3

This state complex shows the "*Moving to Intermediate Target*" states.

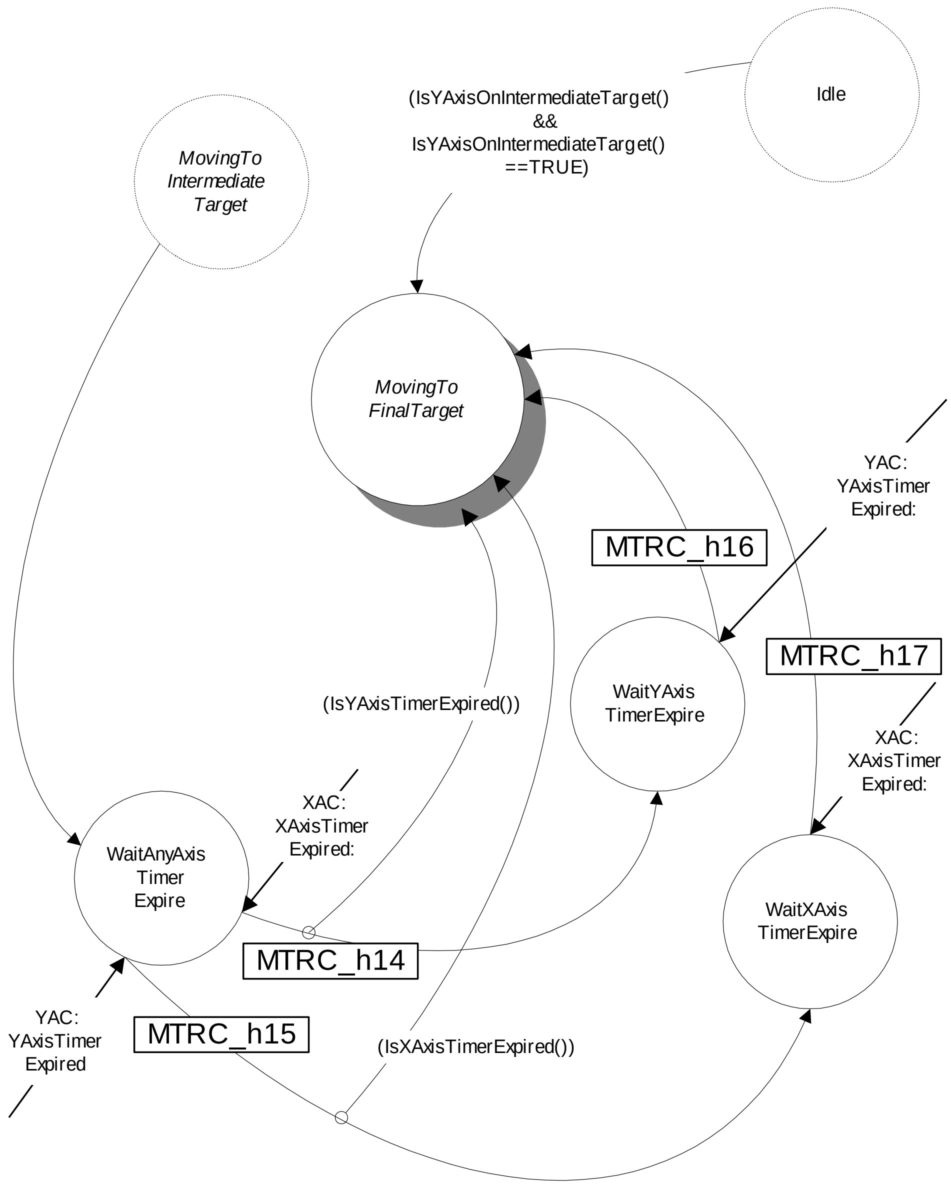
- MTRC enters the "WaitAnyAxisOnIntermediateTarget" only when a move needs to be made. Before the machine enters this state, a move command is sent for the appropriate axis.
- The backlash distance is stored by the Table Parameters Data Manager
- If both axis are already on target, the machine goes directly from Idle to the "Moving to Final Target" states (See next page)
- It is possible to get a "TimeOut" message from one of the axis timers while in one of the "WaitAxisOnIntermediateTarget" states. We ignore this message and check the status of the timer, expired or not, to determine the next step. (Also shown in the next page.



### 6.11.1.2.5. MTRC – GotoXYTarget, State Transitions (Backlash Compensation Enabled) – Level 3

This state complex shows the "*Waiting For Axis Timers To Expire*" states. If one of the axis timers have already expired, then it immediately goes specifically into one of the 2 "Wait Timer Expire States". Otherwise, if both axis timers have not yet expired, then it goes to a state where it waits for any of the two to expire.

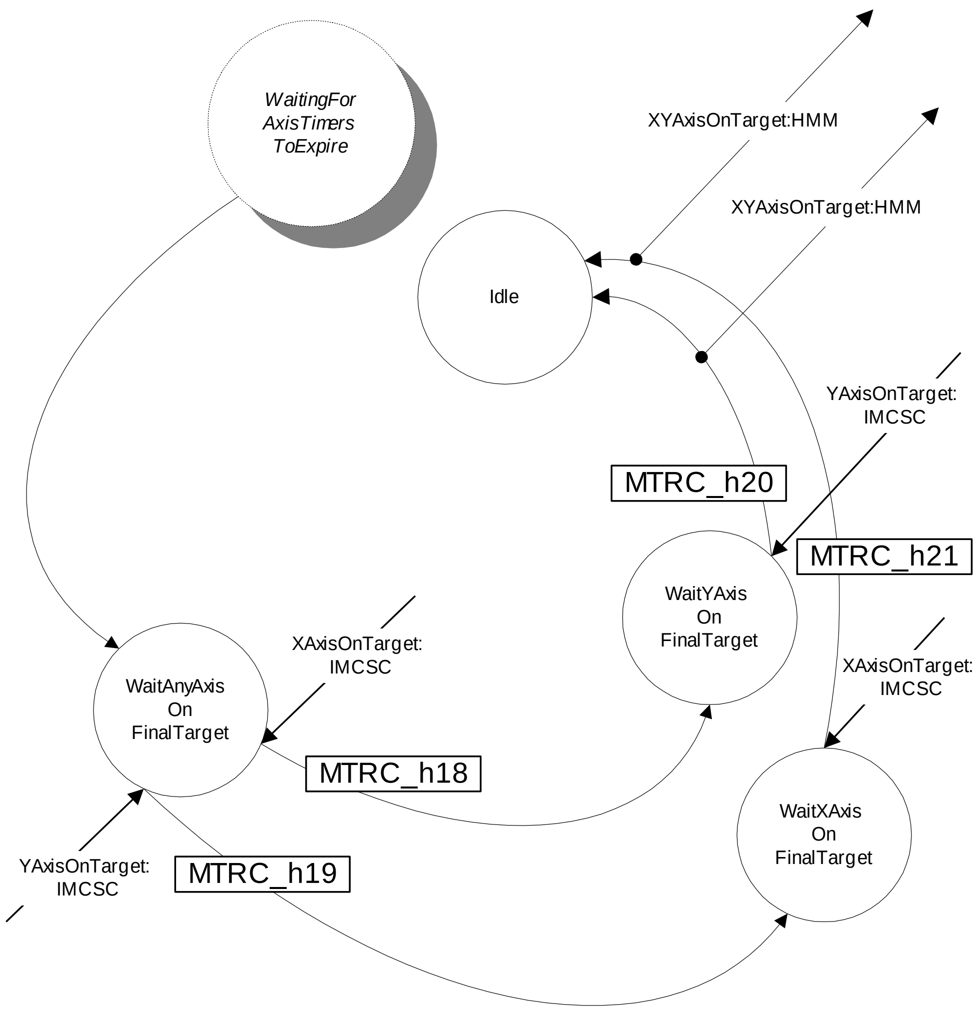
Note: If both axis are already on their intermediate target, the machine goes directly from "Idle" straight to the "Moving to Final Target" states.



### 6.11.1.2.6. MTRC – GotoXYTarget, State Transitions (Backlash Compensation Enabled) – Level 3

This state complex shows the "*Moving to Final Target*" states.

- Before the machine enters the "WaitAnyAxisOnFinalTarget" state, a move command would have already been sent previously by the preceding state exit procedures.
- HMM is notified about the completion of the request before the machine goes back to idle.
- Having an intermediate position implies that both axis will have to move to get to their final target. The diagram below shows that the machine always waits for the other axis to finish when one is done moving



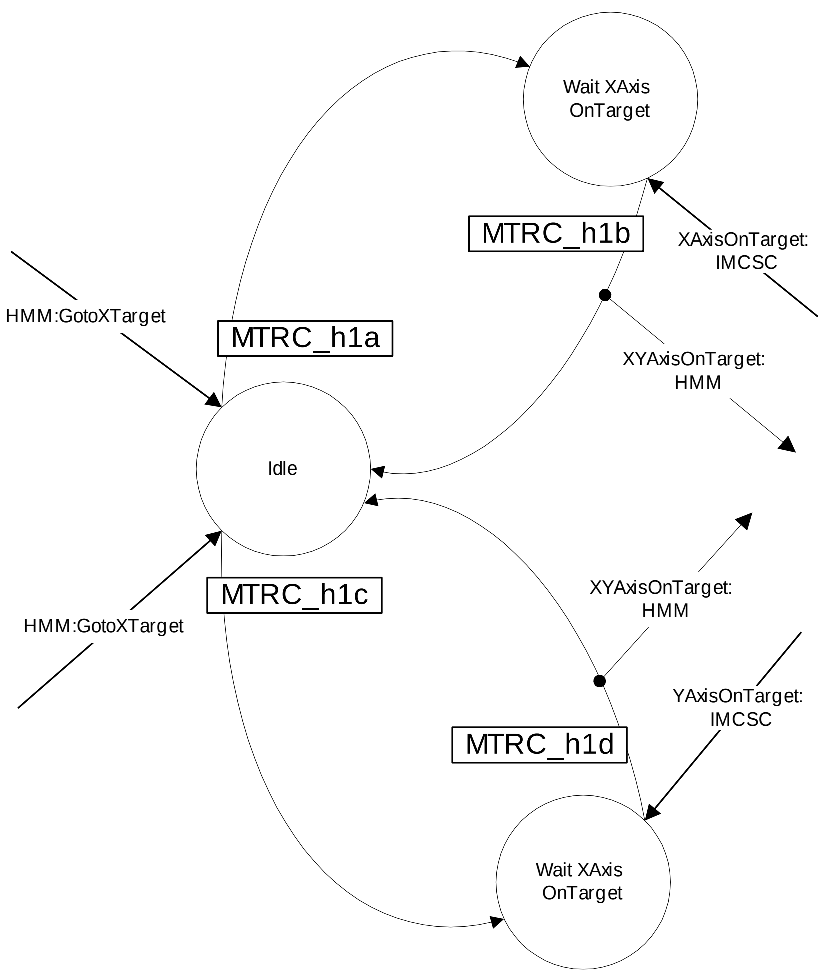
### 6.11.1.2.7. MTRC – GotoXTarget, GotoYTarget State Transitions

These messages allow the moving of only one axis. When a "Goto Target" command is received, MTRC sends the appropriate move command to the motor chip set to start performing the move. Once the move is complete, MTRC receives an "Axis On Target" event. It then informs HMM about the completion of the request.

The "Goto Target" message is accompanied with data.

Data1 = Target Coordinate (X or Y)
Data2 = NULL

NOTE: Backlash compensation is always disabled when performing these requests.



6.11.1.2.1. MTRC – SetNewMoveProfile, State Transitions

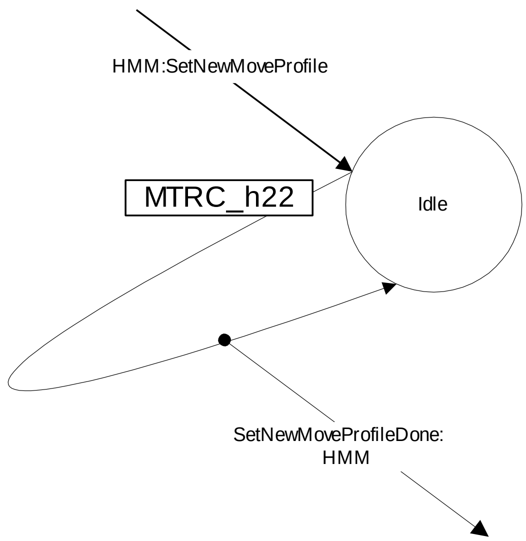
**IMPORTANT:**

**MTRC does not change the move profile by itself. The task of changing the move profiles is given to HMM who knows when to use all the different move profiles.**

The "SetNewMoveProfile" is a message sent by HMM. This message is accompanied with data:

Data1 = Index To The Move Profile Table
Data2 = NULL

MTRC retrieves the move profile data from the Table Parameters Data Manager using the given index.



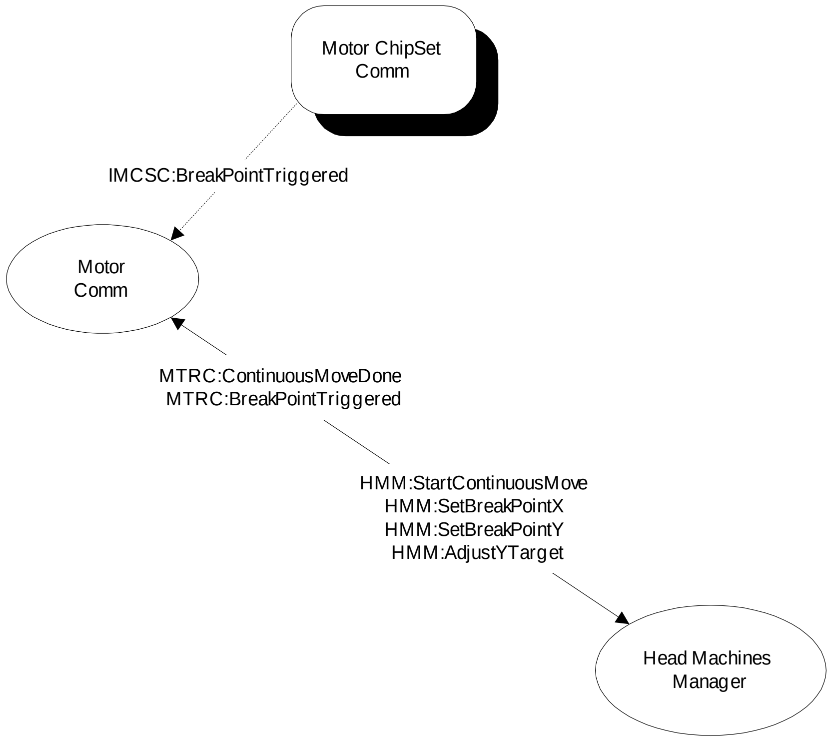

Note:

The changing of the move profiles can only be done while MTRC is in the idle state.

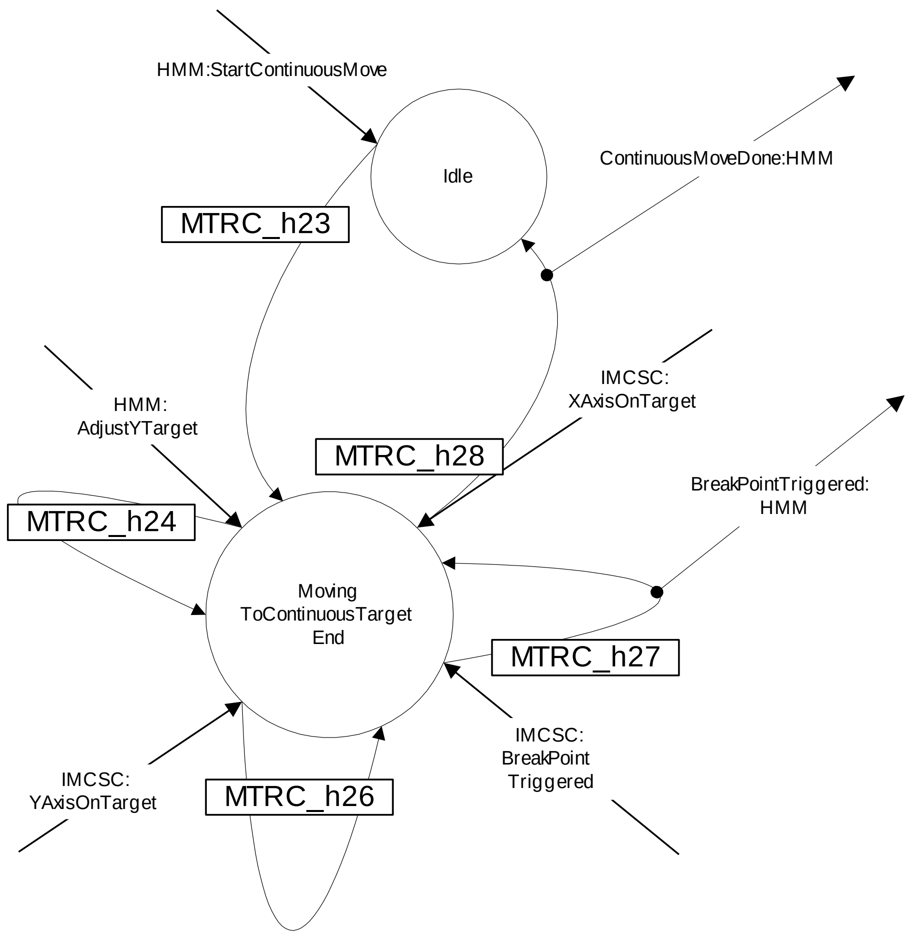
### 6.11.1.3. MTRC – System Interaction, StartContinuousMove

The diagram below shows the components involved.

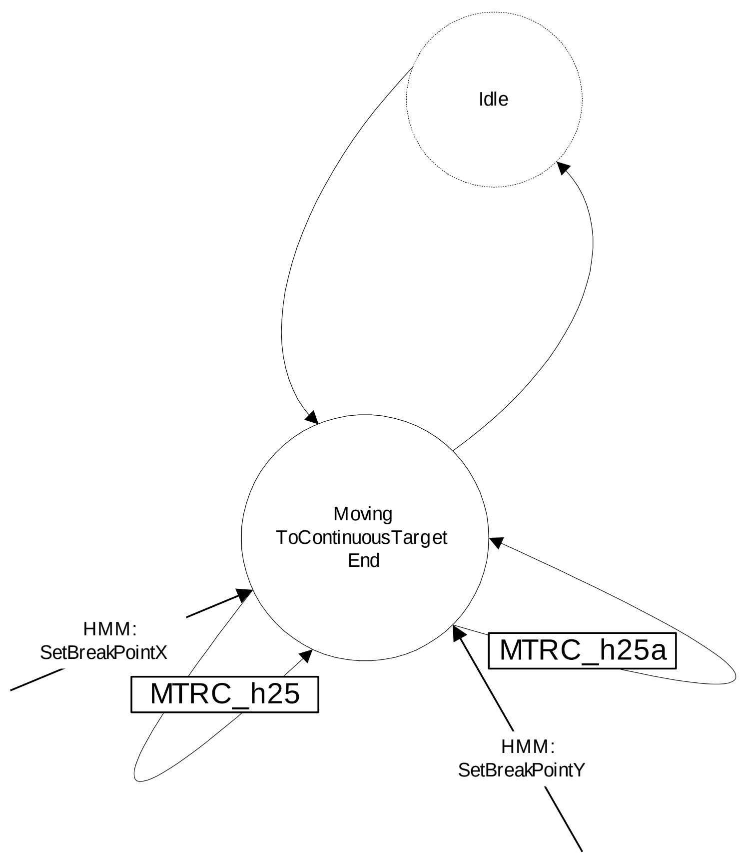
HMM generates this request when he is performing Color Bar Scans or Alignment Scans.



## 6.11.1.3.1. MTRC – Start Continuous Move State Transitions 1



### 6.11.1.3.1. MTRC – Start Continuous Move State Transitions 2



### 6.11.1.4. X Axis Timer (XAT) Task Description

***XAT – X A**xis **T**imer*

As stated earlier, MTRC requires special timing delays when it is performing backlash compensation operations. XAT provides all the necessary timing services for the X axis operation.

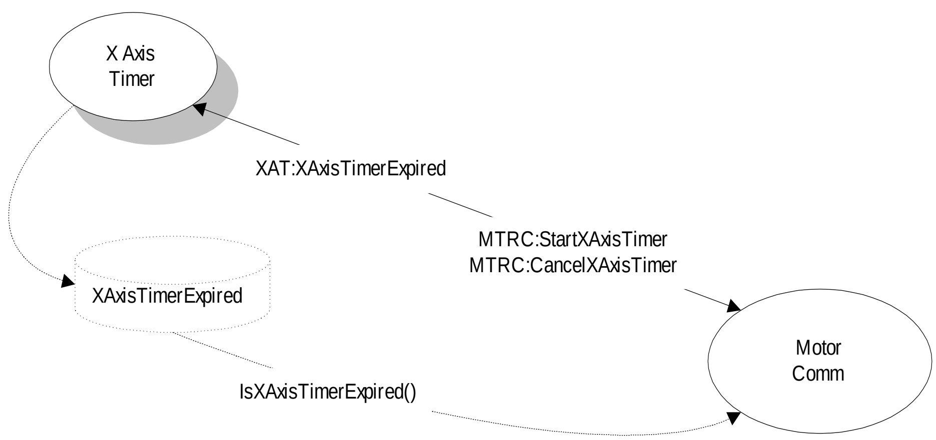
The “StartXAxisTimer” is a message request sent by MTRC. This message is accompanied with data:

Data1 = Timeout Duration in milliseconds
Data2 = NULL

**IMPORTANT:**

**The timeout requested is subject to the Kernel’s timer limitations.**

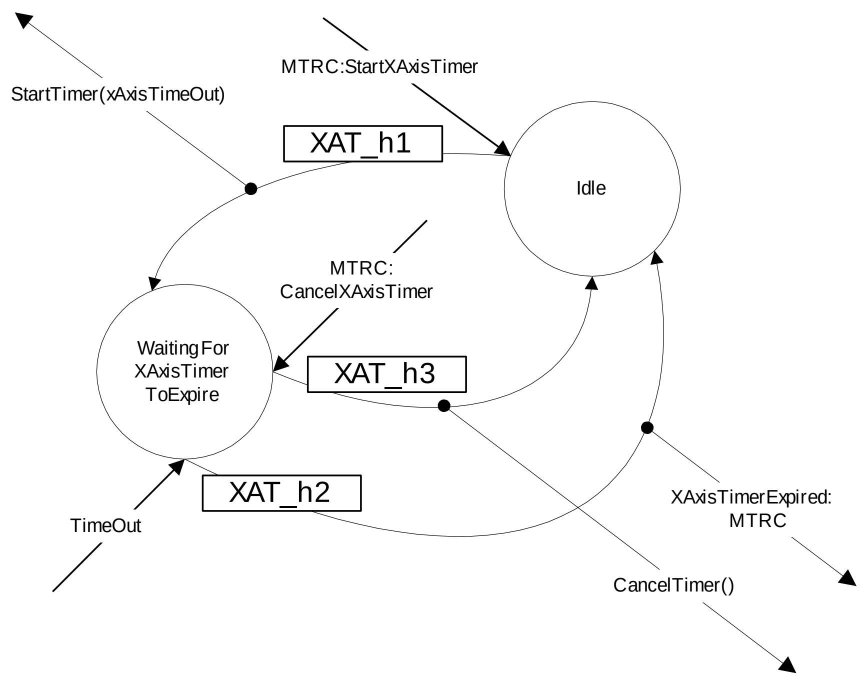
### 6.11.1.5. XAC – System Interaction, StartXAxisTimer, CancelXAxisTimer



### 6.11.1.5.1. XAT – StartXAxisTimer, State Transitions

XAT only has a few simple state transitions. When it receives a "StartXAxisTimer" message, it generates a Timer Request to the Kernel and then goes to a "Waiting" state. The Kernel sends a TimeOut message when the requested time has expired. XAT in turn tells MTRC about the event.

Note: The timer request can also be cancelled.



### 6.11.1.6. Y Axis Timer (YAT) Task Description

***YAT – Y Axis Timer***

YAT performs the same services as that of XAT. The only difference is that these services are used for Y-axis operations instead.

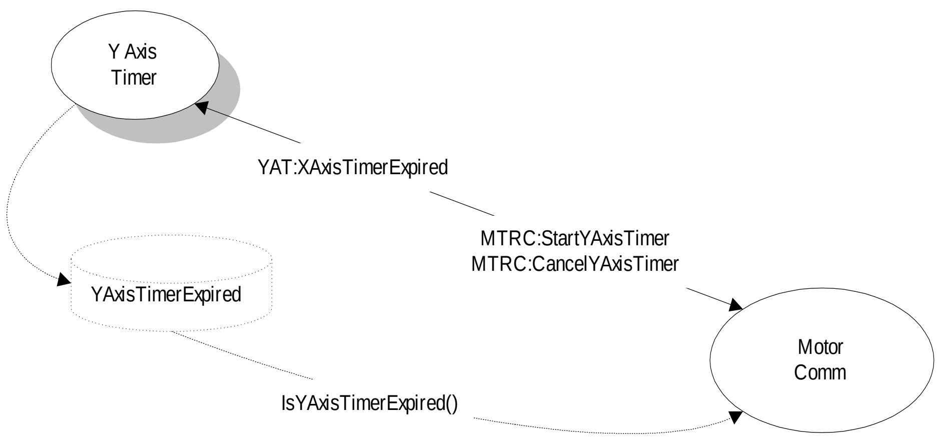
The "StartYAxisTimer" is a message sent by MTRC. This message is accompanied with data:

Data1 = Timeout Duration in milliseconds
Data2 = NULL

**IMPORTANT:**

**The timeout requested is subject to the Kernel's timer limitations.**

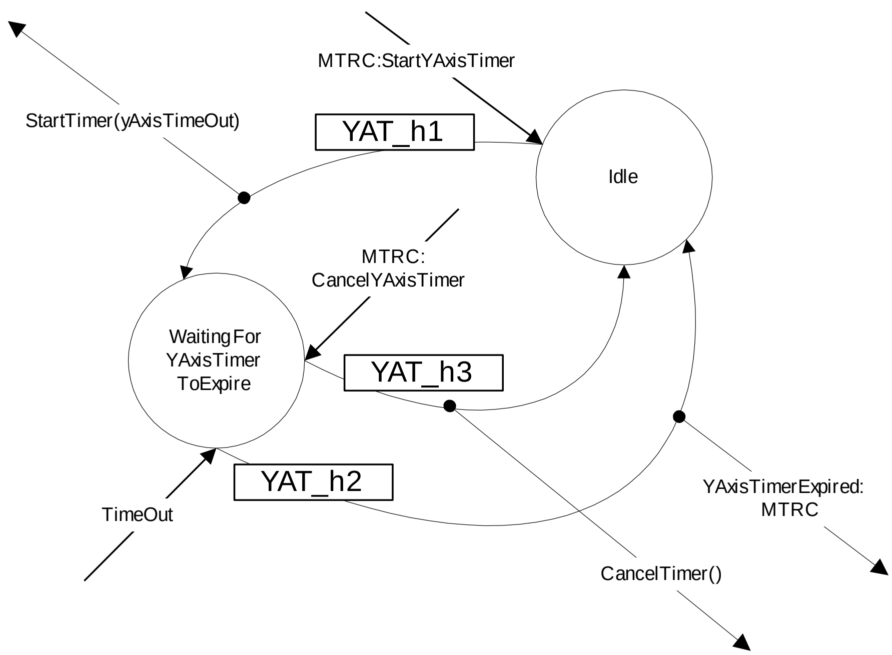
### 6.11.1.7. YAT – System Interaction, StartYAxisTimer, CancelYAxisTimer



### 6.11.1.7.1. YAT – StartYAxisTimer, State Transitions

YAT only has a few simple state transitions. When it receives a "StartYAxisTimer" message, it generates a Timer Request to the Kernel and then goes to a "Waiting" state. The Kernel sends a TimeOut message when the requested time has expired. YAT in turn tells MTRC about the event.

Note: The timer request can also be cancelled.



### 6.11.1.8. Target Lamp Manager (TLM) Task Description

***TLM – T****arget* ***L****amp* ***M****anager*

To avoid premature failures, the target lamp needs to be automatically turned off after some given time. This SM keeps track of the amount of time that the Target Lamp has been turned or how long CSM has been active (probe head not moving). The starting time reference is the last time a "Target Lamp On" request was received from CSM.

Note: TLM will be needed by CSM when he is in any of the 3 modes below:

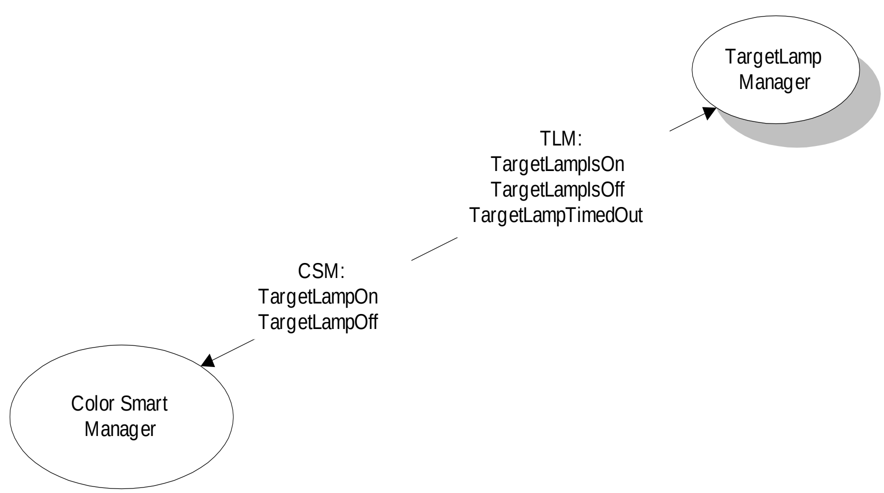
1. Scanning Mode
2. Track Ball Mode
3. Development Diagnostic Mode

**IMPORTANT:** **TLM does not automatically turn the target lamp off himself. He notifies CSM of the timeouts and CSM in turn decides if a Target Lamp Off request should be made.**

**To maintain synchronization of the Head State Machine complex, CSM should be in a wait state immediately after making a "Target Lamp On/Off" request to TLM.**

**CSM is in charge of re-sending the "Target Lamp On" request when necessary that is when table activity resumes. This way the Target Lamp can be re-enabled after it has been automatically turned off by TLM.**

## 6.11.2. TLM – High Level Diagram

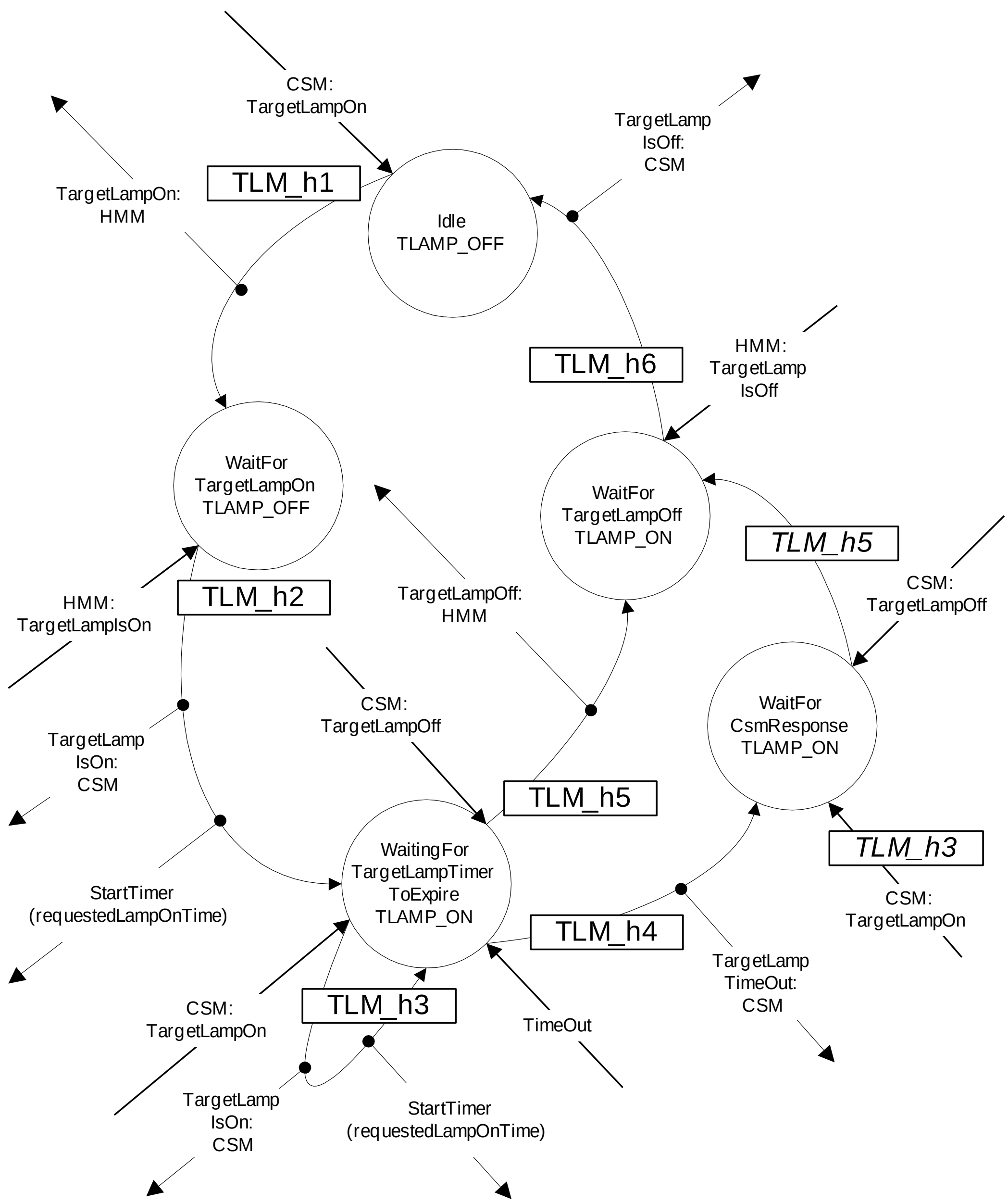


### 6.11.2.1.1. TLM – TargetLampOn, TargetLampOff, State Transitions

The “Target Lamp On” Message is direct request form CSM and is accompanied by data: The TLAMP_ON/OFF indicates the current status of the Target Lamp within the different states.

Data 1 = the Target Lamp Timeout

Note: A “Flash Effect” can be achieved by giving Data 1 a small value.



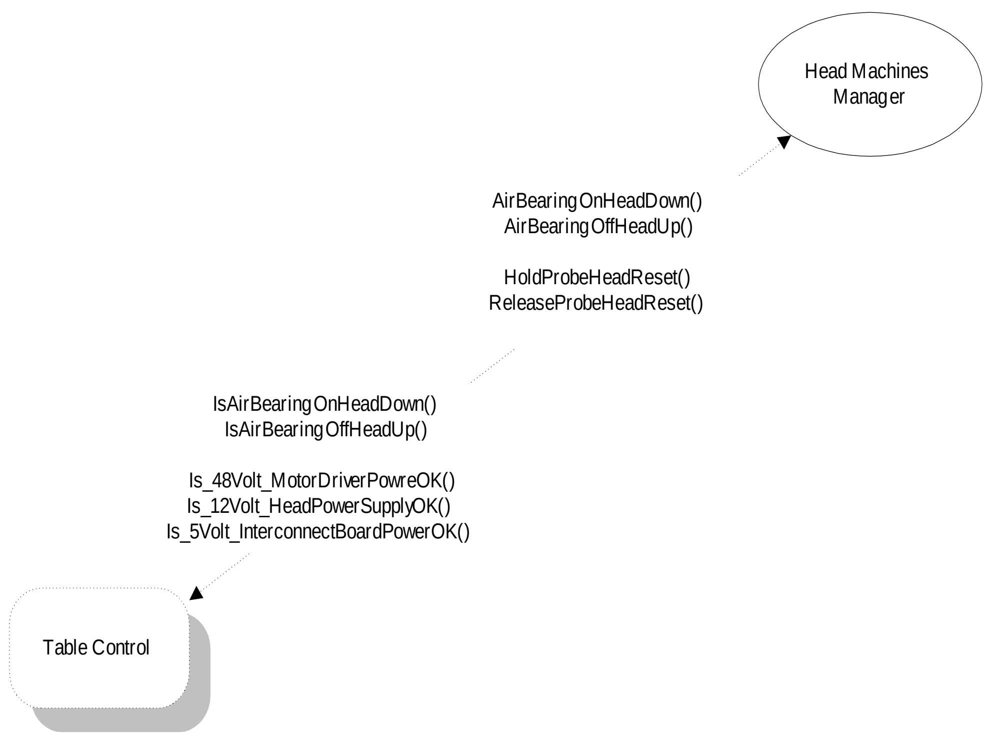
## *6.12. Table Control (TC) Task Description*

### ***TC - T**able **C**ontrol*

In general these are just a set of functions or interface to allow the table air to be turned on or off. A separate RESET line has been provided to force the head to reset itself. TC also provides diagnostics services.

Since the Head Machines Manager handles TC, its operation will be synchronized with the rest of the system.

### 6.12.1. TC – High Level Context Diagram



## *6.13. Head Operations Coordinator (HOC) Task Description*

### ***HOC** – **H**ead **O**perations **C**oordinator*

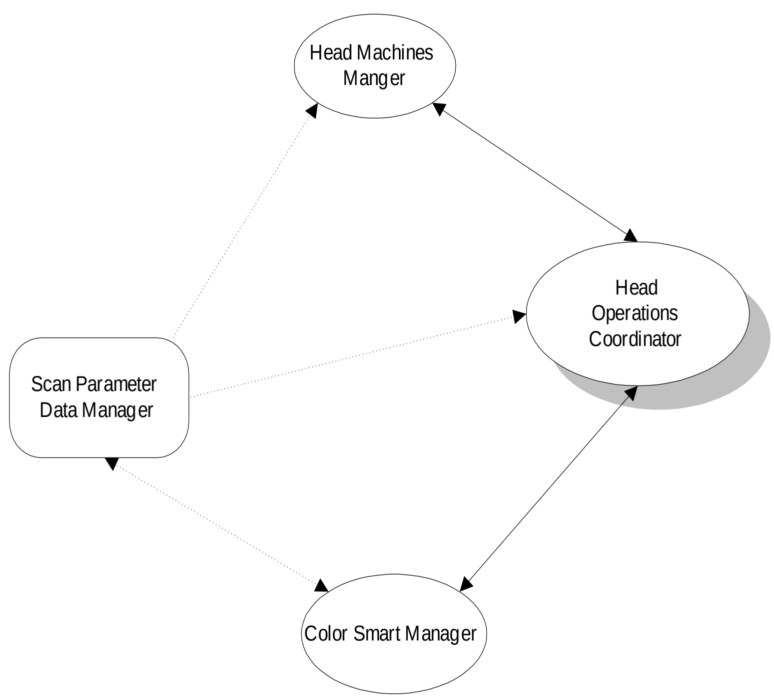
This SM was created to simplify the state transitions of both HMM and CSM.

- He can send Color Bar or a Point To Point Scan command to HMM
- He also knows how to perform a Full Calibration procedure
- He knows the appropriate time to lower or raise the probe head (both HMM and CSM do not have this intelligence)
- He knows when to apply the backlash compensation, as in the case of the alignment scans
- Any measurement operation should be channeled through HOC. HOC will be ready for all of the different measurement requests

This SM is in charge of sending the different type of scan commands to HMM. He has direct access to the Scan Parameter Data so he can determine the type of scan to be performed. The two types of scan are PointToPoint Scan or ColorBar Scan.

### 6.13.1. HOC - High Level Context Diagram

As seen below, HOC interacts with many other machines within the system. The diagram below shows the machines as well as the messages that are passed between them.

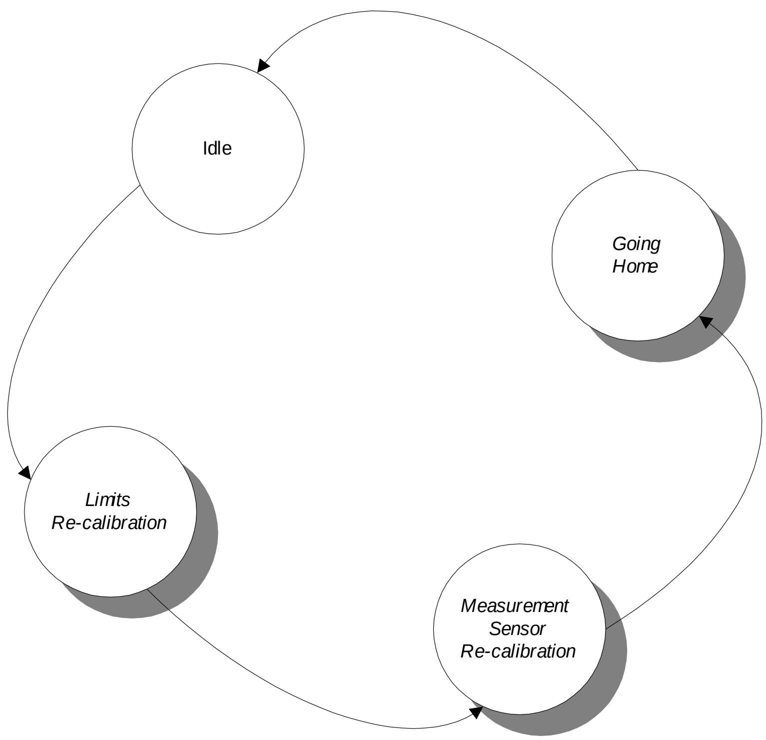


#### 6.13.1.1.1. HOC – Do Full Calibration, State Transitions Level 1

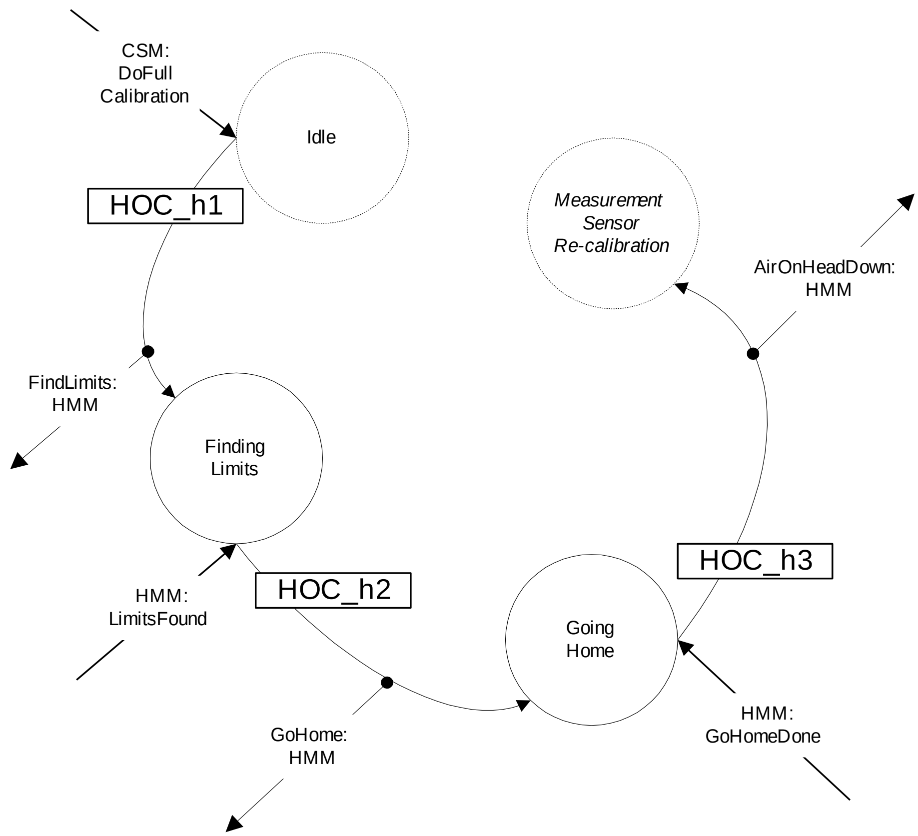
The “Do Full Calibration” message is generated directly by CSM.

There are many sequences involved when performing this operation. This is why the different states are further shown at another level.

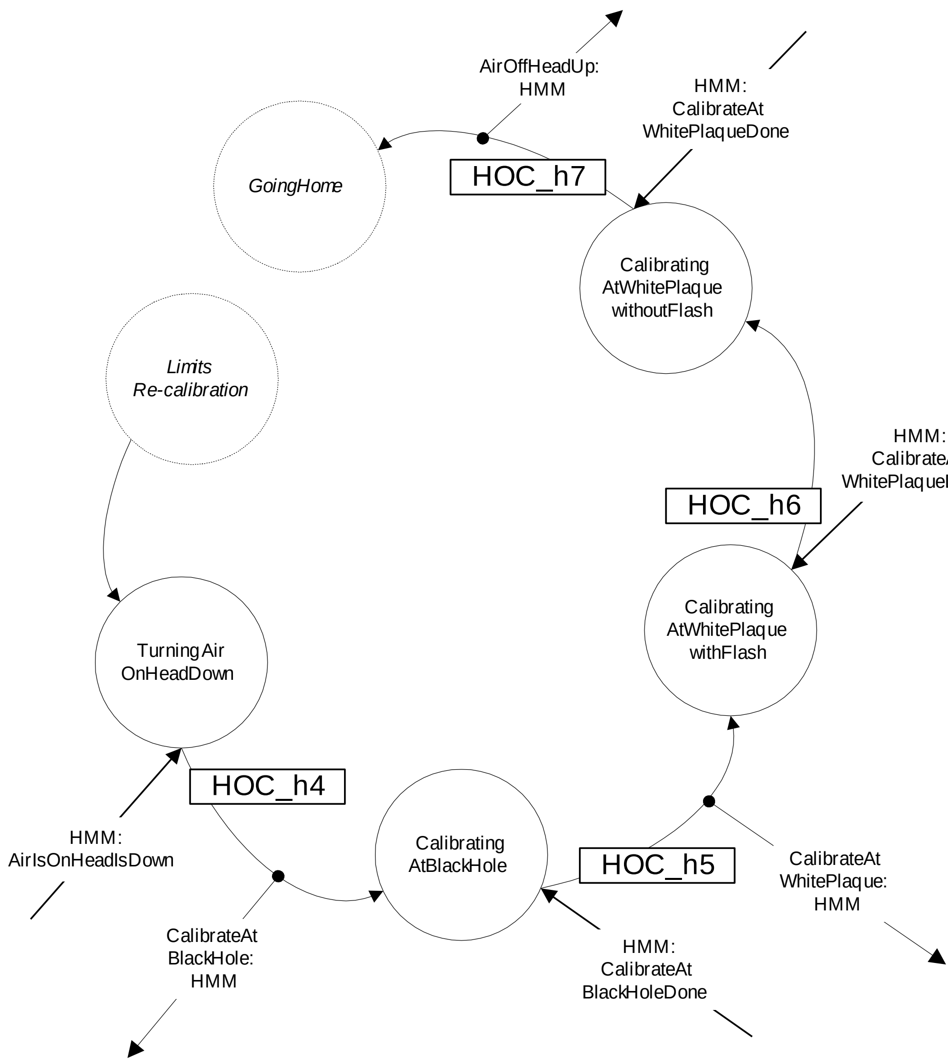
1. The limits are re-calibrated to maintain positioning accuracy
2. The reference patches are re-measured to maintain measurement accuracy
3. And finally the probe head is parked a the home position



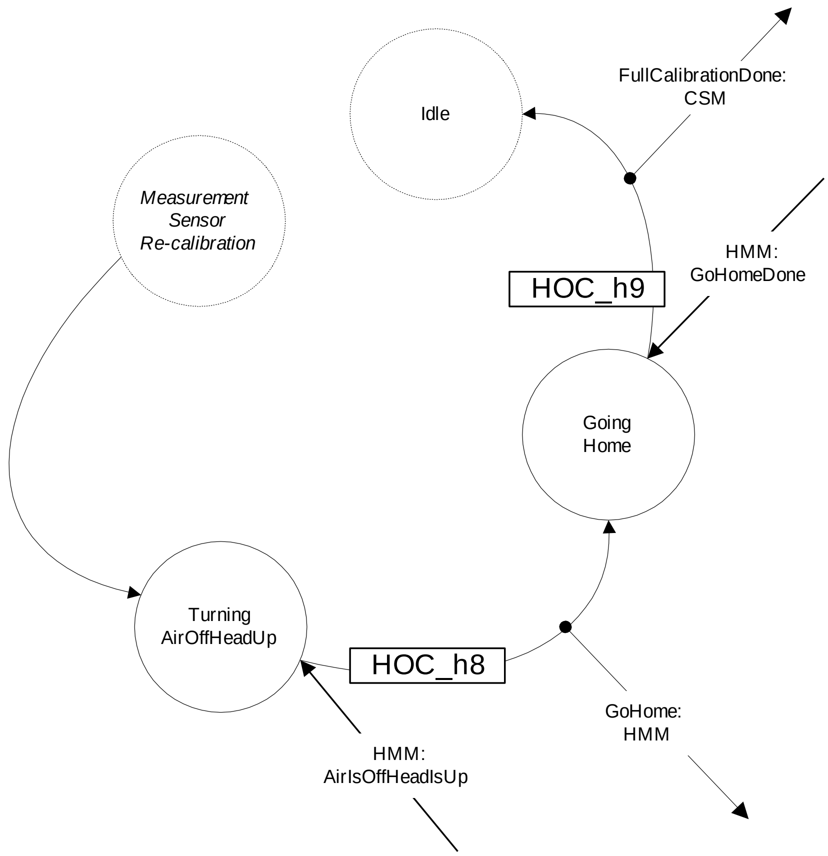
### 6.13.1.1.2. HOC –*Full Calibration, Limits Re-Calibration*, State Transitions Level 2



### 6.13.1.1.3. HOC –Full Calibration, Measurement Sensor Re-Calibration, State Transitions Level 2



### 6.13.1.1.4. HOC –*Full Calibration Going Home*, State Transitions Level 2

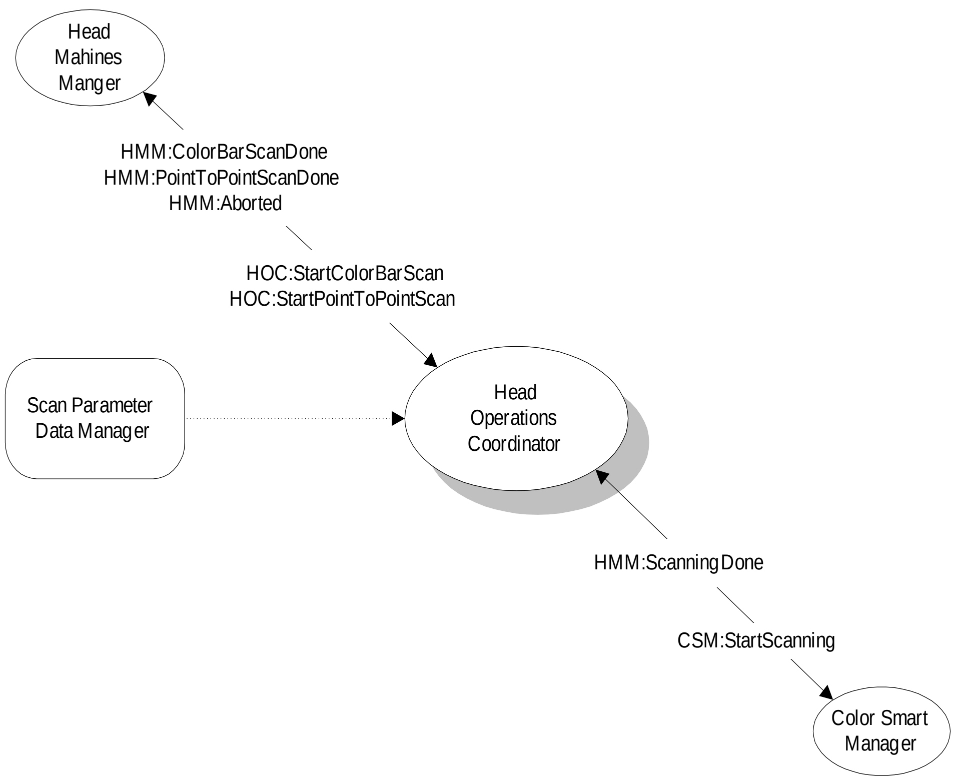


### 6.13.1.2. HOC – System Interaction, StartScanning

The Start Scanning is a generic message from CSM.

It is assumed that the NT Application has performed a Set Scan Parameter process before this request is made. HOC deciphers the scan parameters to determine whether a series of scan parameters should be performed as point to point or as color bar scans. HOC goes through the entire scan parameter data until all the points have been measured.

HOC has the ability to report the percentage done to CSM.



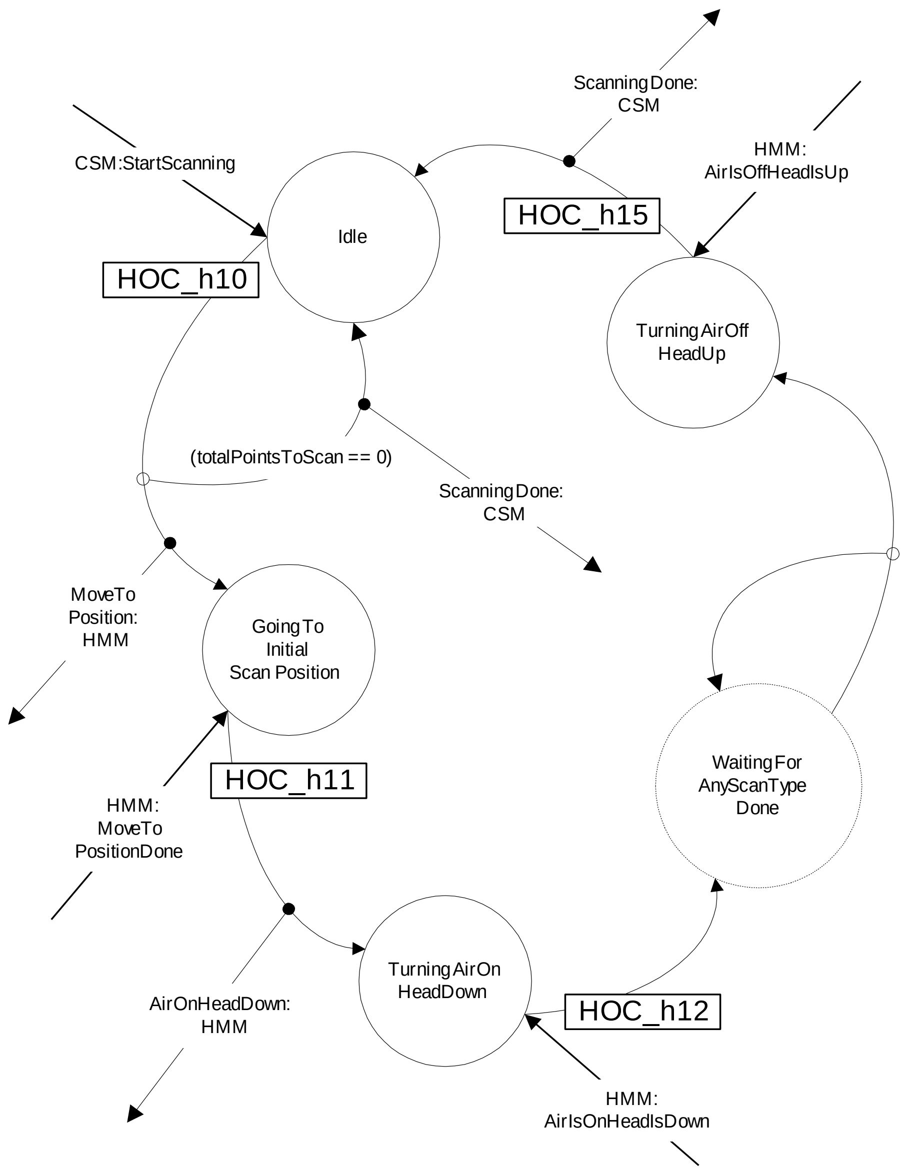
### 6.13.1.2.1. HOC – Start Scanning, State Transitions Level 1

The “Start Scanning” message is generated directly by CSM.

This assumes that CSM is in the correct mode and that it has performed the calibration procedures – Measurement Calibrations and Limit Calibrations.

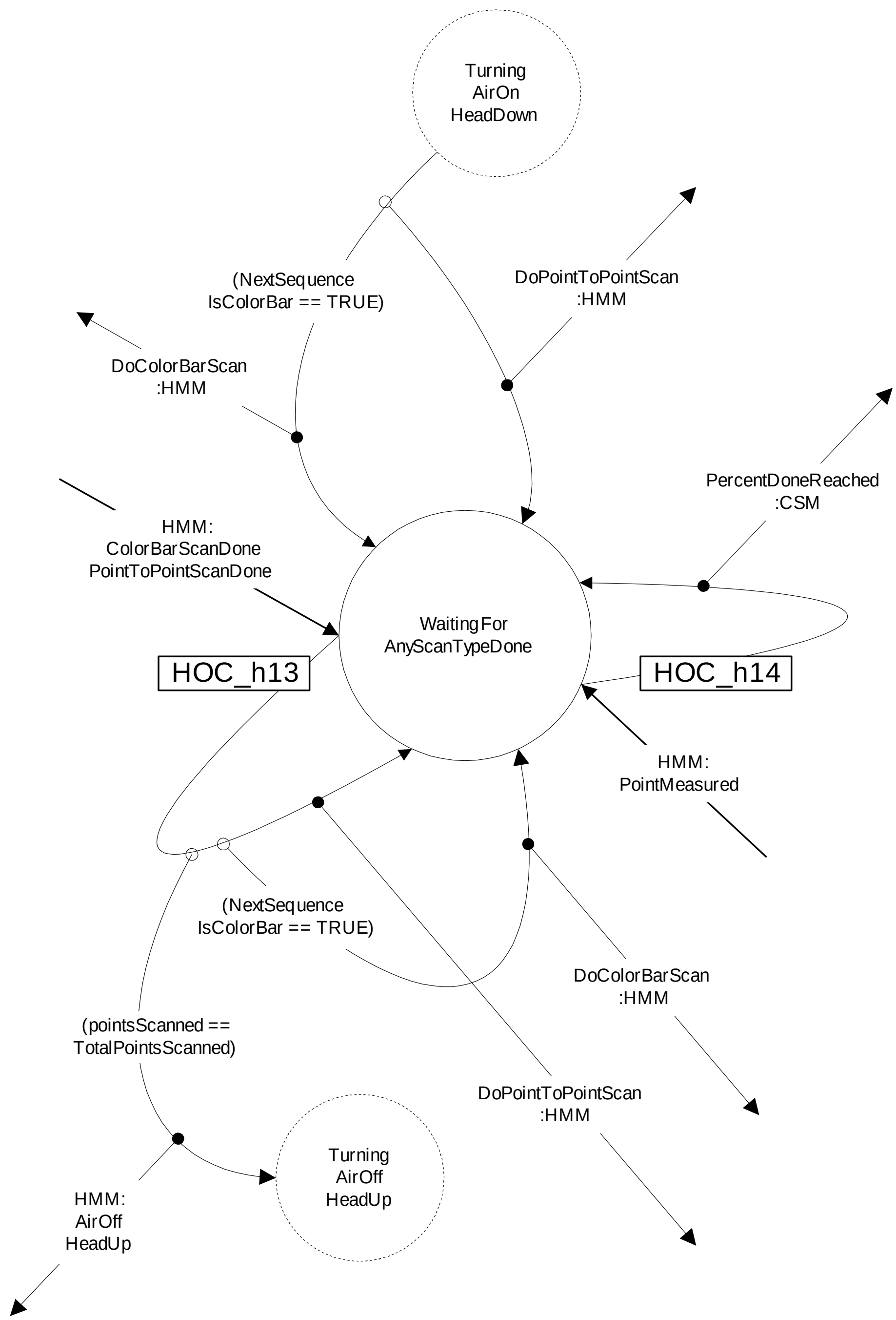
Data 1 = report percentage done flag – 0 = No Report
1 = Report

Data 2 = percentage mark (0% – 100%)



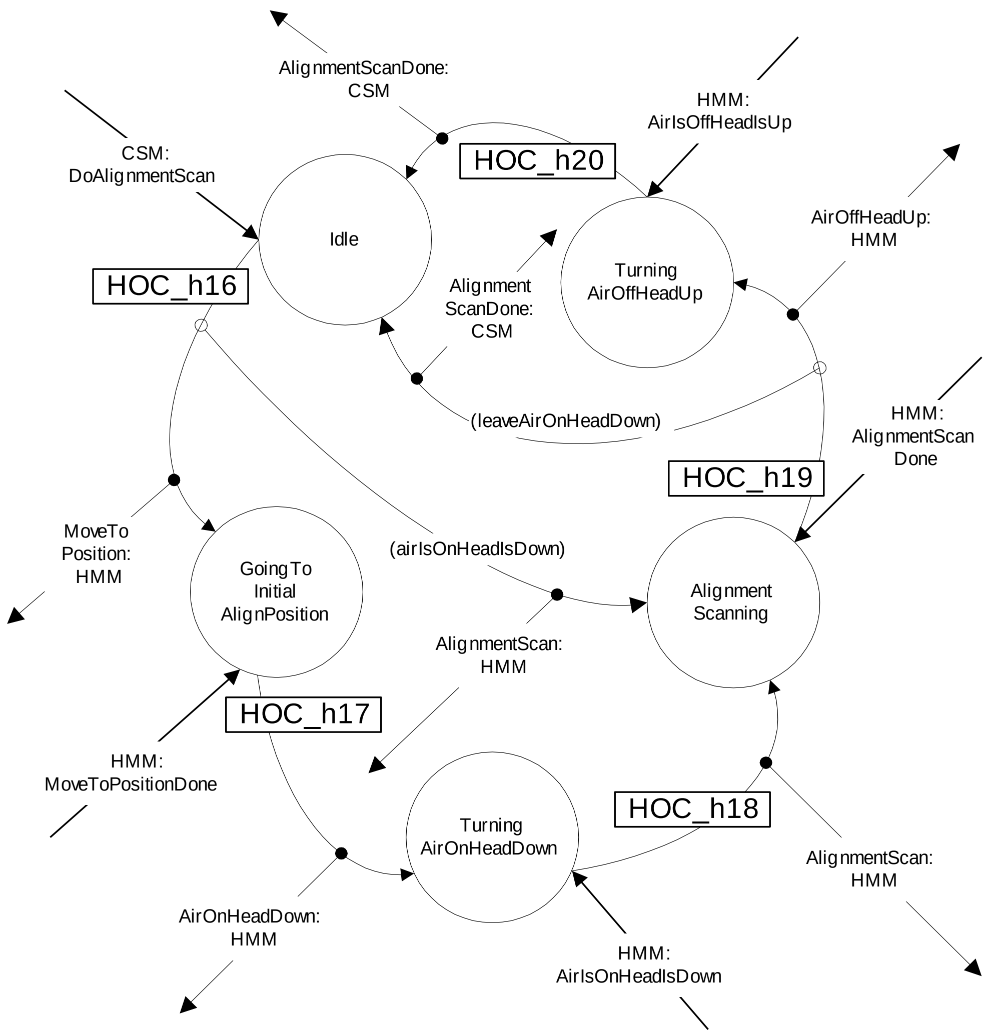
### 6.13.1.2.2. HOC – Waiting For Scan Selected, State Transitions Level 2

The “Start Scanning” message is generated directly by CSM.



### 6.13.1.2.3. HOC – Do Alignment Scan, State Transitions

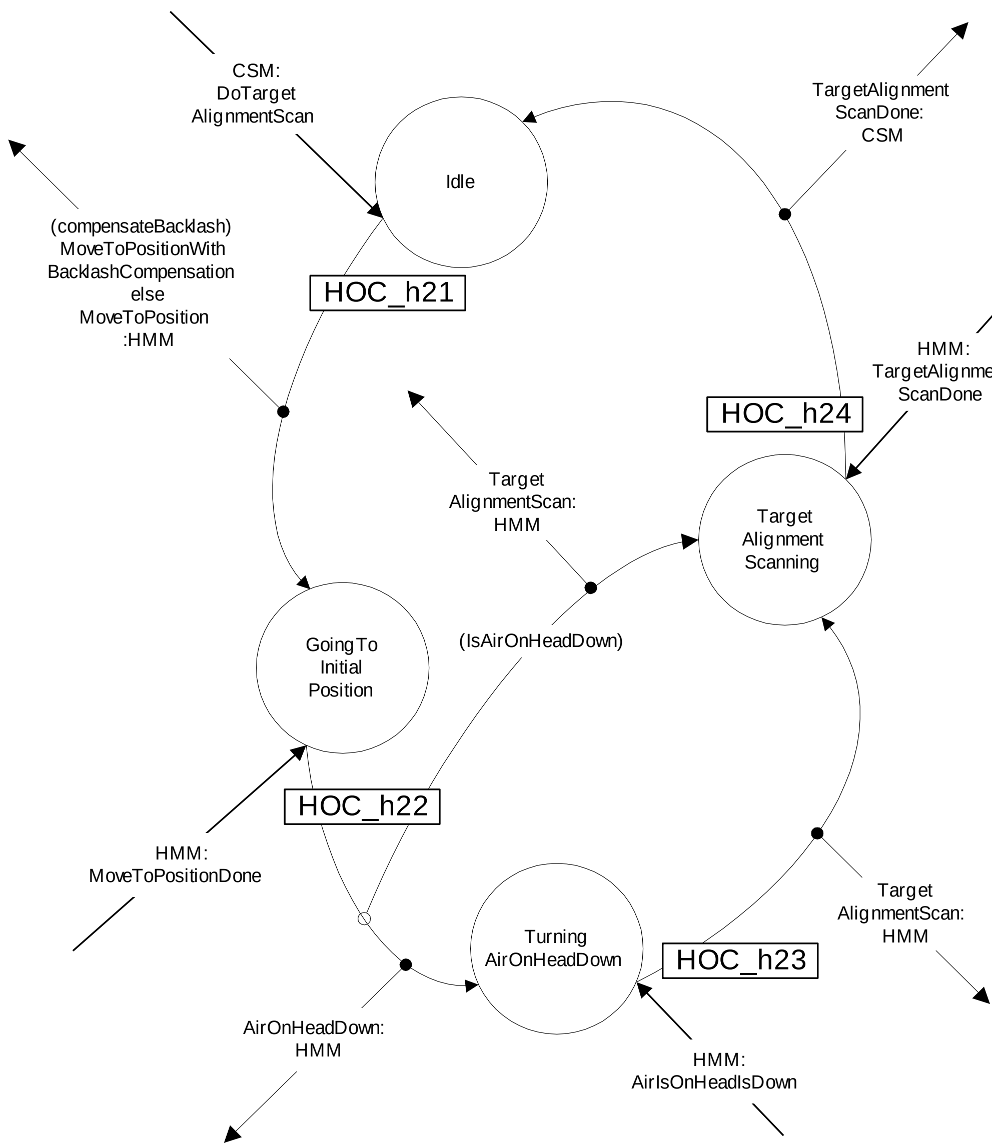
The “Alignment Scan” message is generated directly by CSM.



### 6.13.1.2.4. HOC – Do Target Alignment Scan, State Transitions

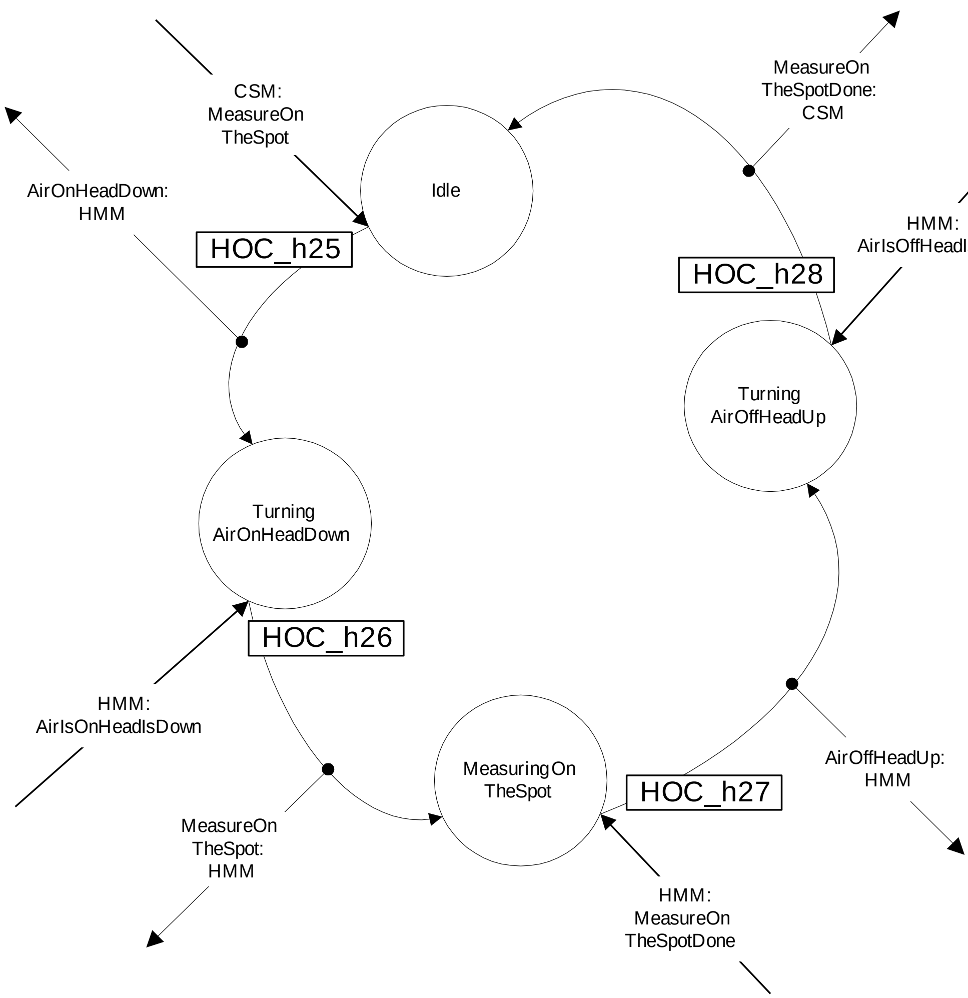
The NT Application will make 6 of these requests in order to perform the Target Alignment procedure. The "Target Alignment Scan" message is generated directly by CSM.

**IMPORTANT: As seen in the diagram below, once the air is turned on it remains on. CSM will receive a message from the NT Application to indicate that the Target Alignment procedure is over. This consequently causes CSM to deactivate the air and raise the probe head.**



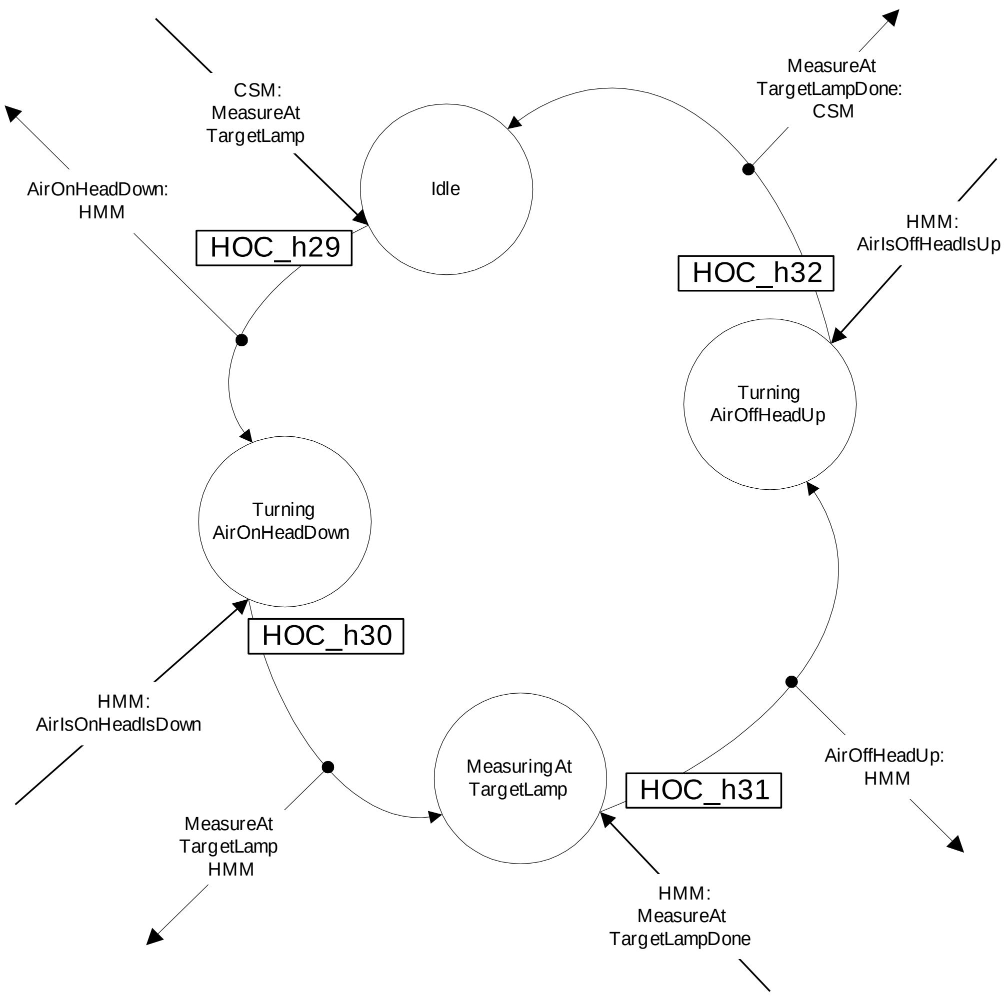
### 6.13.1.2.5. HOC – Measure On The Spot, State Transitions

The “Measure On The Spot” message is generated directly by CSM.



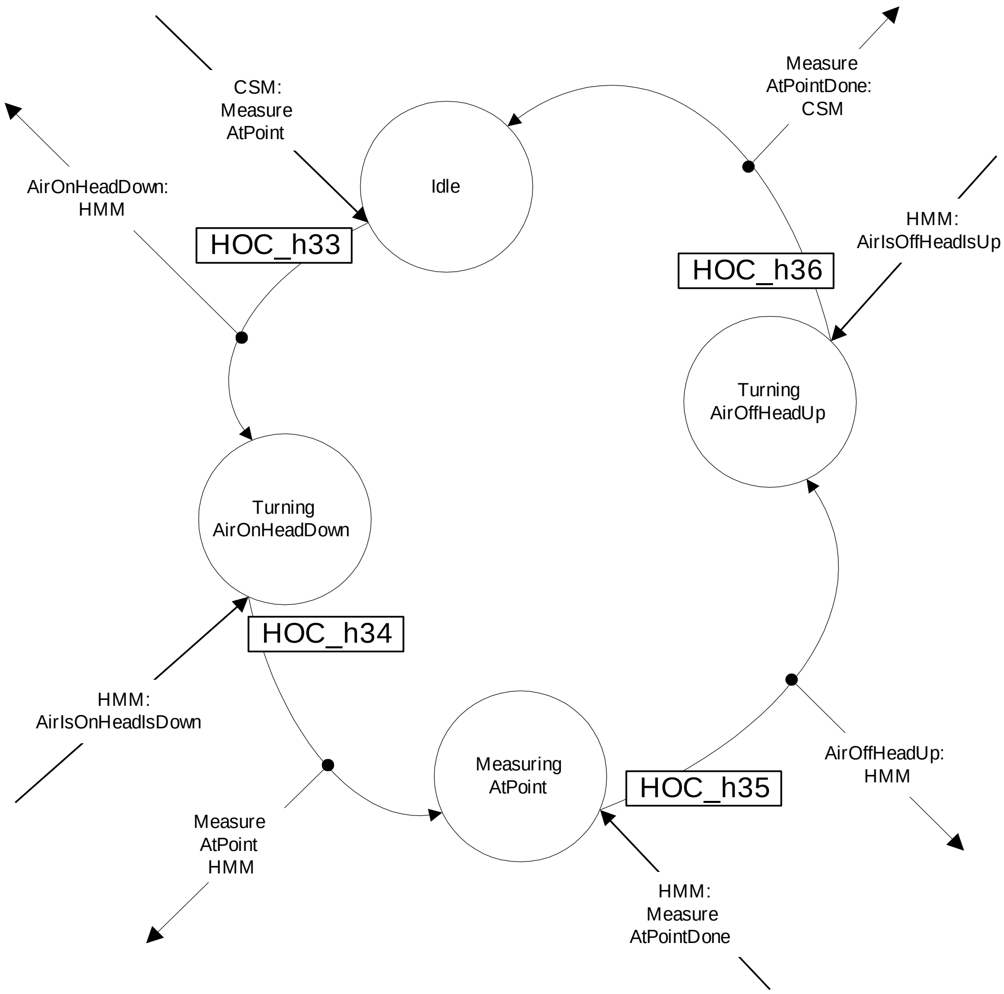
### 6.13.1.2.6. HOC – Measure At Target Lamp, State Transitions

The “Measure At Target Lamp” message is generated directly by CSM.



### 6.13.1.2.7. HOC – Measure At Point, State Transitions

The “Measure At Point” message is generated directly by CSM.



## *6.14. Head Machines Manager (HMM) Task Description*

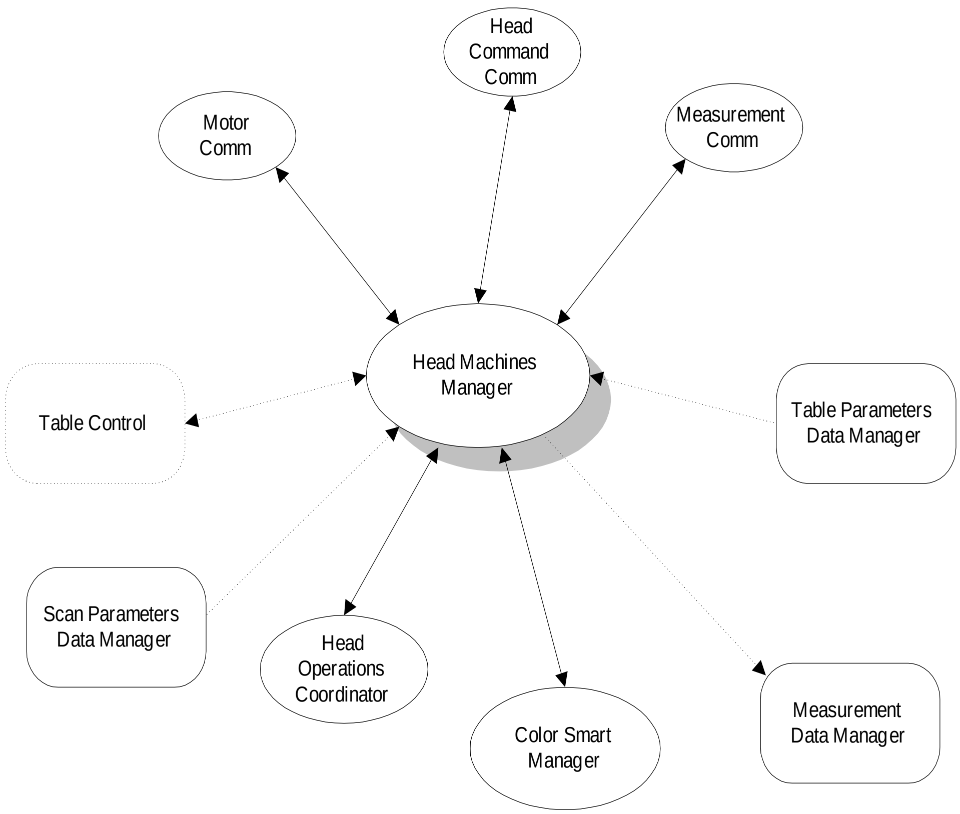
### ***HMM - H**ead **M**achines **M**anager*

As the name suggests, this SM has the task of managing all of the SM used for head operations. This includes coordinating and synchronizing head operations during a scan:

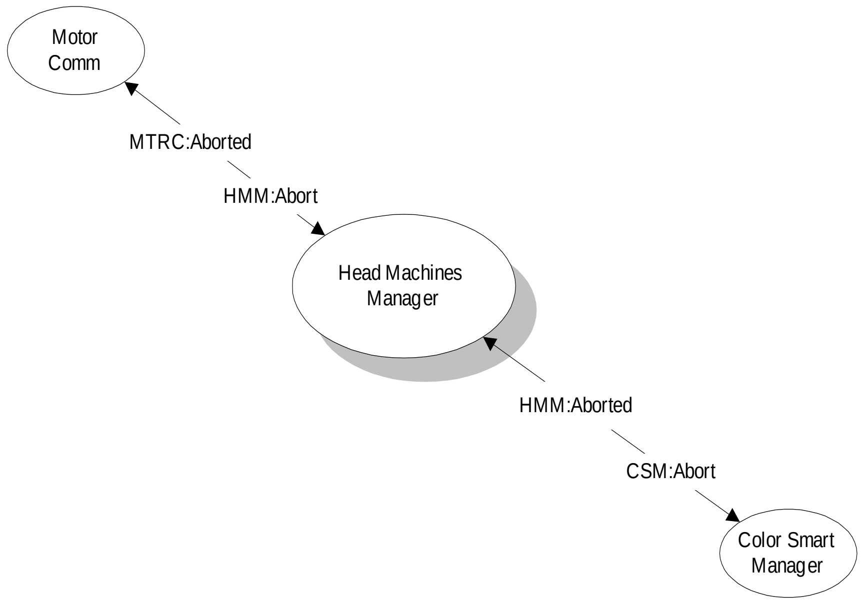
- He knows the general state of the head, including its position in the table
- We will allow short commands from Color Smart NT App such as “Turn On/Off Light”. Such a command will be execute if appropriate.
- He will only respond to a Color Smart NT App command if the command is appropriate for the current state. For instance a “Turn On/Off Head Air” will be processed and ignored if the head is currently in the process of scanning (as an example)
- He generally accepts commands from Color Smart NT App as relayed by the Color Smart Manager.
- He receives direct commands from the Head Operations Coordinator.
- He has the intelligence and knows how the different types of scan are performed. He has control over several subordinate tasks, which is necessary to allow such a coordinated operation
- He can turn on/off the table air when appropriate
- To query the head target positions, he is provided with a direct connection with SPDM who manages the SCAN PARAMETER data.
- He knows what move profiles to use and when. In some cases, CSM chooses the move profiles to be used

## 6.14.1. HMM - High Level Context Diagram

As seen below, HMM interacts with many other machines within the system. The diagram below shows the machines as well as the messages that are passed between them.

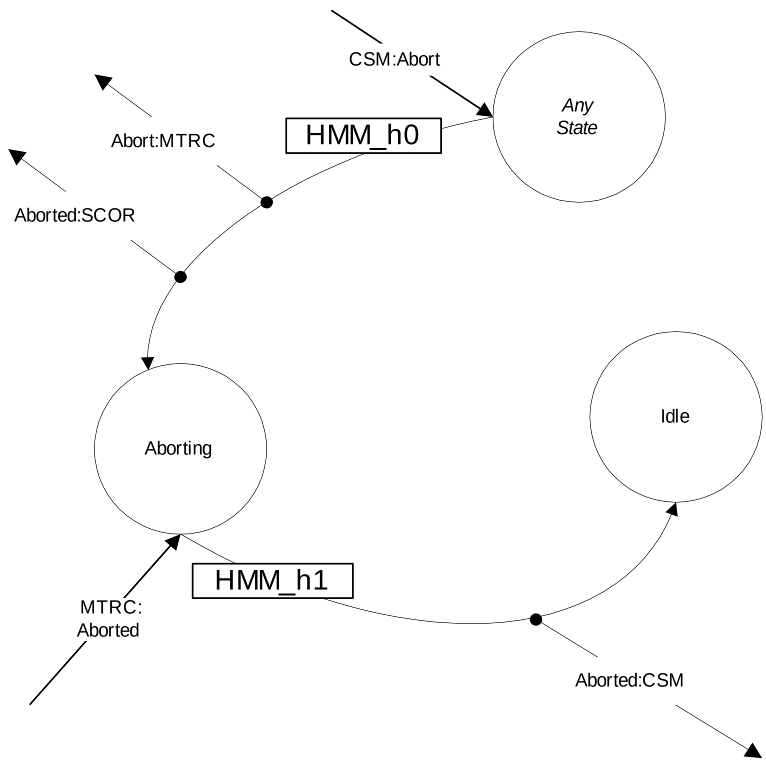


### 6.14.1.1. HMM – System Interaction, Abort

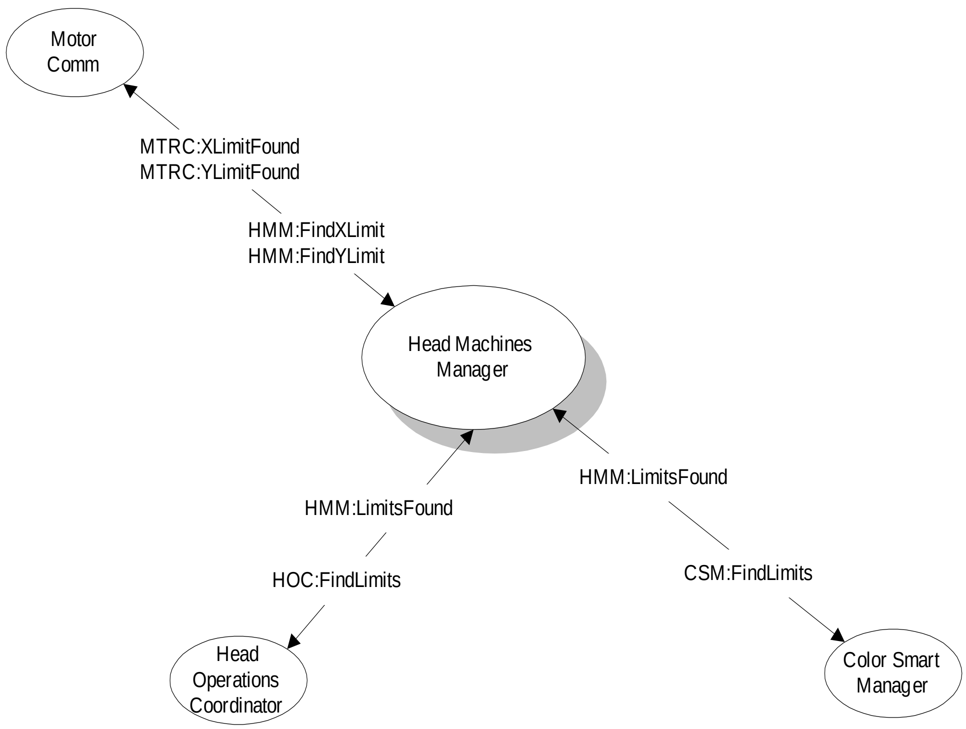


### 6.14.1.1.1. HMM – Abort, State Transitions

The “Abort” message is generated directly by CSM. HOC is notified just in case a scan was in progress.



## 6.14.1.2. HMM – System Interaction, Find Limits



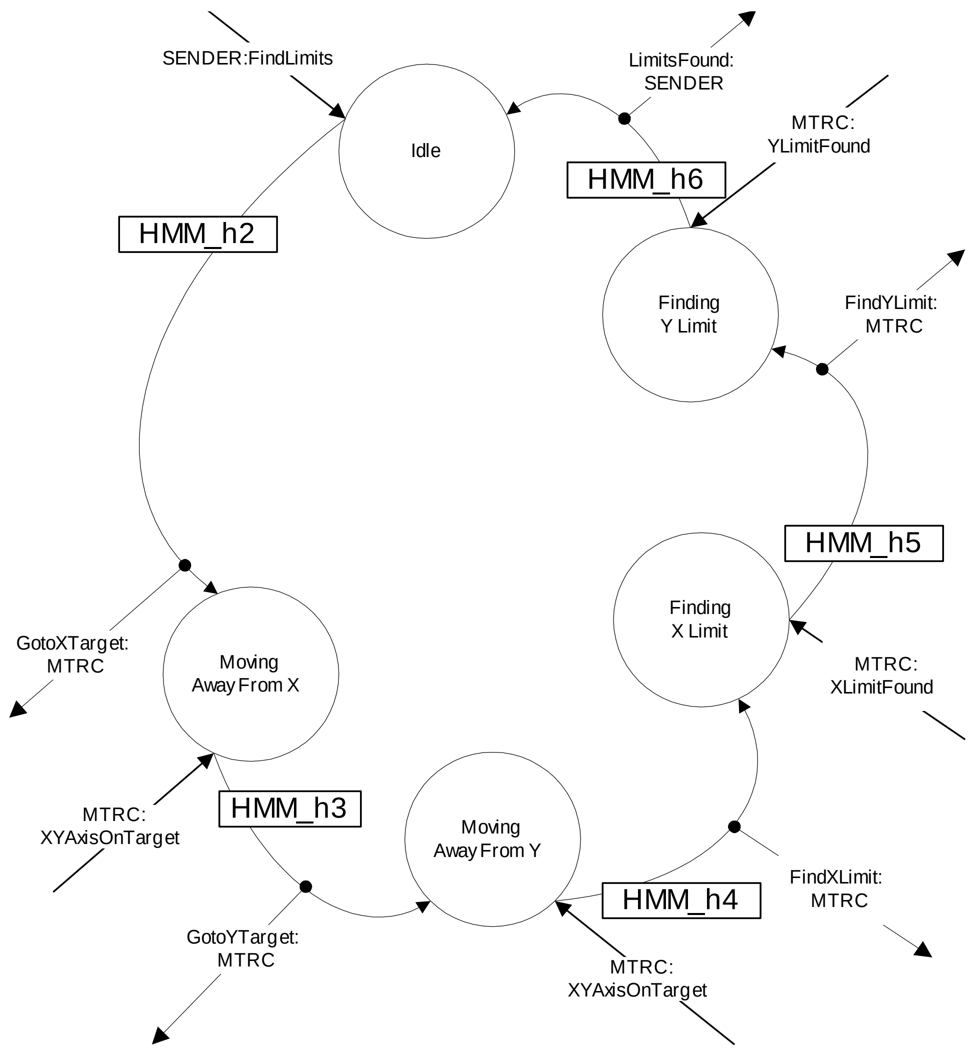
### 6.14.1.2.1. HMM – Find Limit, State Transitions

CSM or HOC generates the “FindLimit” message.

When the “FindLimit” message is received while in Idle, HMM will send a request to MTRC to move to a given location. This location is some given distance away from the current X location. He then waits until MTRC sends back an “XYAxisOnTarget” message. Once this message is received, he then sends another request to MTRC to also move away from the current Y location.

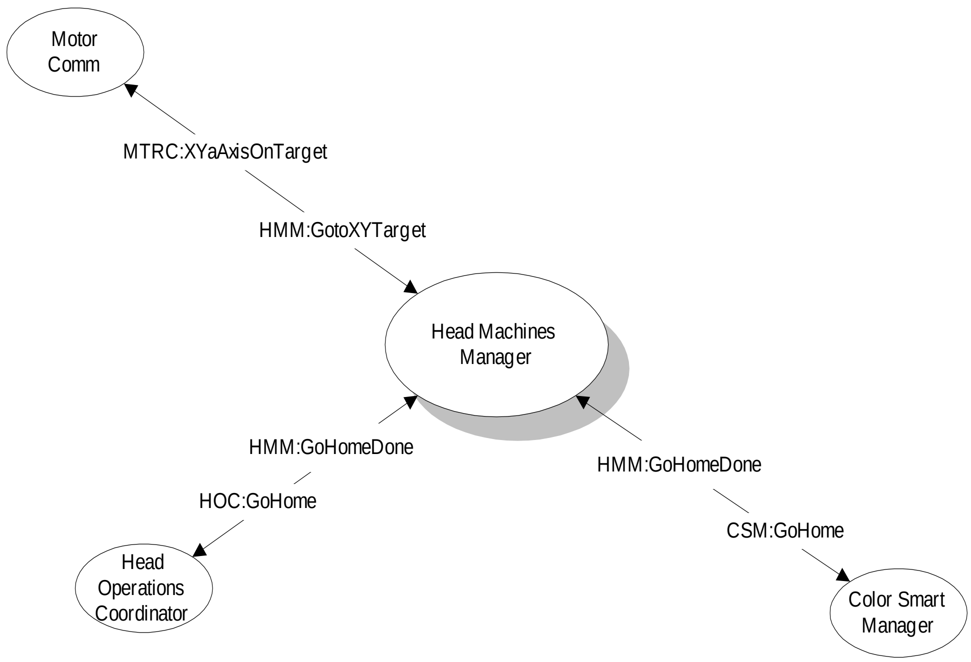
Once these steps are complete a request is sent to MTRC to “FindXLimit”. He waits until MTRC sends back a “XLimitFound” message. We repeat the same process for the Y-axis. Once the Y Limit has been also been found, reply to the SENDER telling him “LimitsFound”. HMM then returns back to the Idle State.

Note: It is up to the SENDER to make sure that the air and probe head is in their correct states.



### 6.14.1.3. HMM System Interaction, Go Home

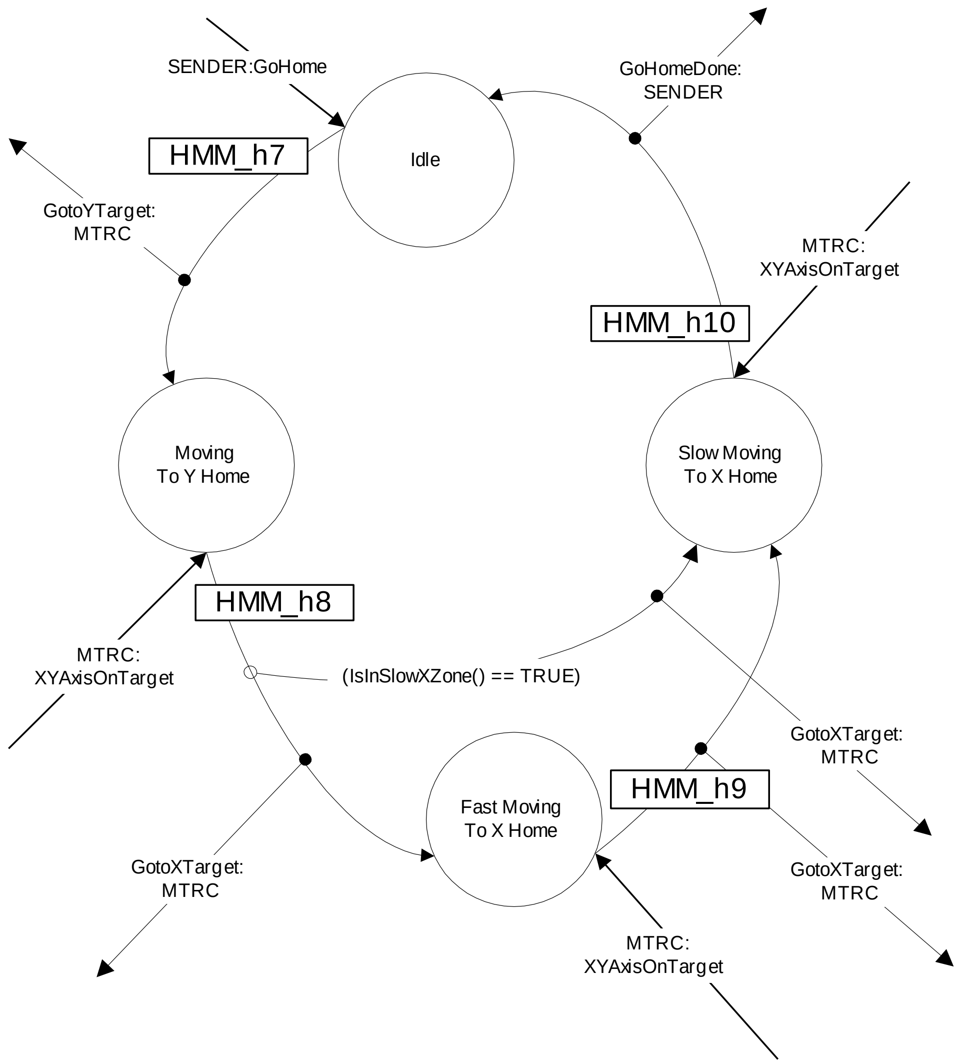
At the end of a scan or the “Trackball” modes, the head is requested to go home and be parked at the white plaque. CSM or HOC directly sends the “Go Home” message. The center of the white plaque is considered to be “Home”. HOC uses this message when it needs to perform a Full Calibration procedure.



### 6.14.1.3.1. HMM – Go Home, State Transitions

Executing a Go Home request requires a special set of moves. When going home, the head moves towards Y home at using one move profile. Then it moves towards X home. The move to X home requires two separate profiles. The first is a Fast Move profile. This move is performed until the head reaches a certain X coordinate. The move profile is then changed to a slower one until it finally reaches X home.

The SENDER is notified that its request was completed.

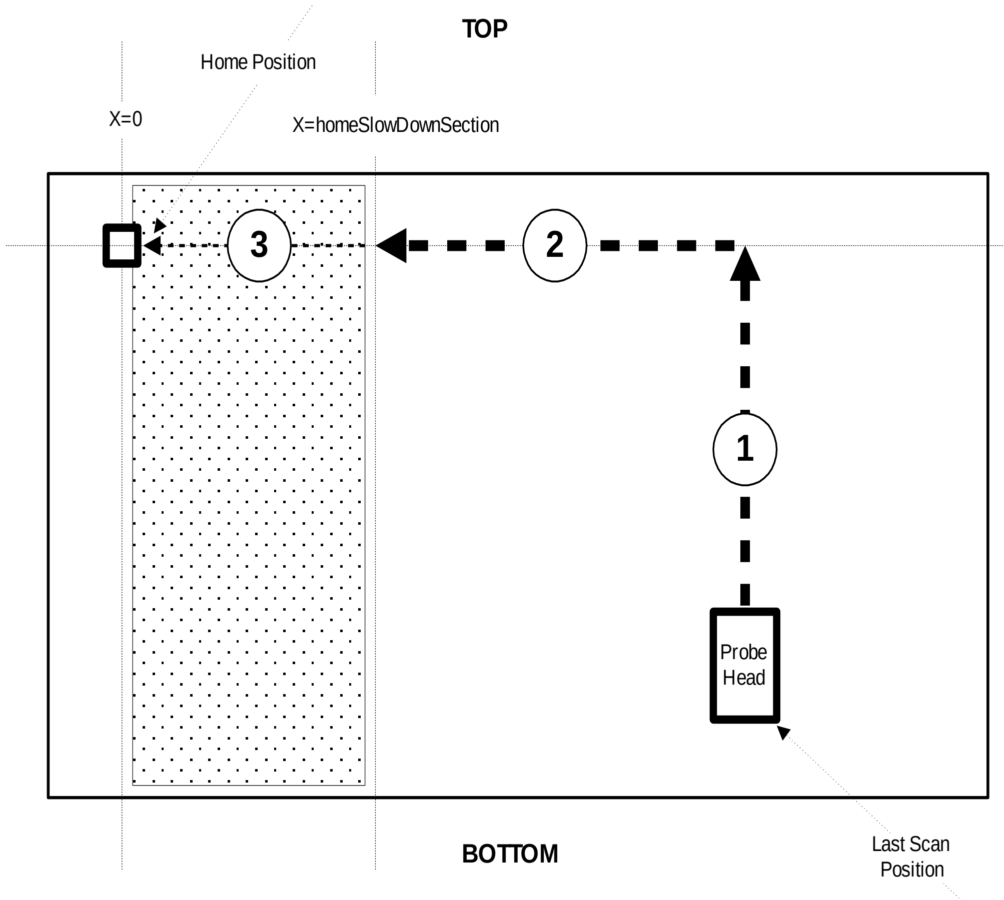



NOTE:

The X and the Y Home coordinates are considered to be the center of the White Plaque.

### 6.14.1.3.2. HMM - Go Home, Mechanical Behavior

If the Probe Head is positioned beyond the gray area, there are 3 different move profiles used when going home, otherwise only 2 move profiles will be needed. After a sheet has been scanned, from a mechanical point of view the Head moves in a fashion shown below:



### 6.14.1.4. HMM System Interaction, Move To Position, Move To Position With Backlash Compensation

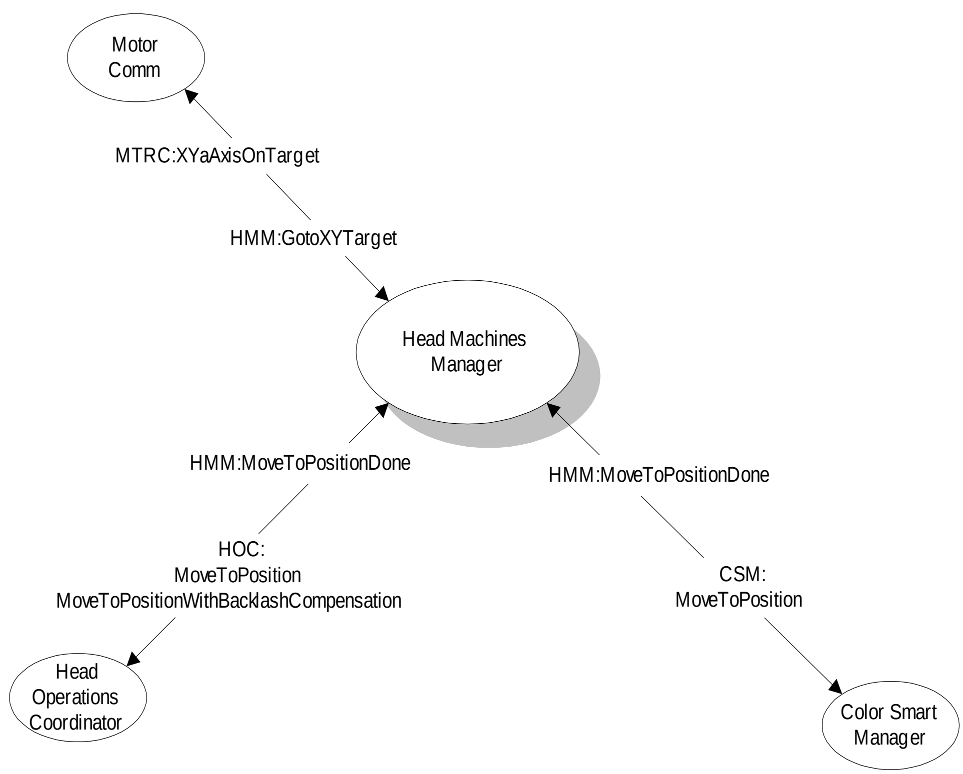
There are 2 types of request that allows the probe head to be positioned.

**Move To Position** - request is normally used by CSM to perform the trackball moves where backlash compensation should be disabled.

**Move To Position With Backlash Compensation** - request is particularly used by HOC to initially position the probe head at the beginning of the Target Alignment Scan. See HOC – Do Target Alignment Scan, State Transitions on page 104 for more information.

**IMPORTANT: This procedure is not used for scanning purposes. Delays implemented in this procedure will not affect the performance of the Point to Point scans.**

The diagram shows all the machines involved in performing the process.



### 6.14.1.4.1. HMM – MoveToPosition, State Transitions

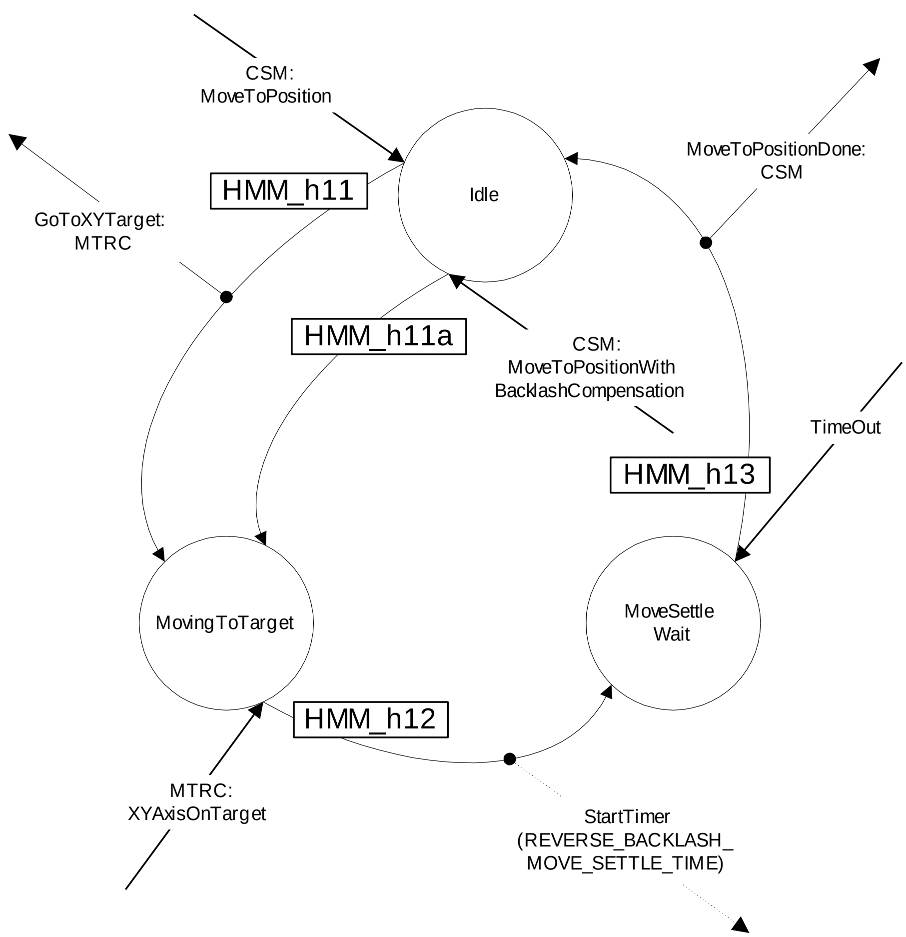
The "Move To Position" message is accompanied by data.

Data1 = X Coordinate
Data2 = Y Coordinate

When HMM receives the message, it forwards the coordinates to MTRC along with the move request. The SENDER is notified once the request has been completed.

Note: HMM always performs a Move Settle Delay for this process. Once HMM is in idle, it can **immediately** process and execute a "Reverse or a Forward Axis Move" command if necessary. This can be the case of a trackball move that causes the step motors to move in the reverse direction with reference to the last move.

The delay time used here is generic and the same as the one used for the Reverse Backlash Move delay. This assumes that the Reverse delay is greater than the Forward delay. See "Timing Parameters Data Access" on page 162 for more information.

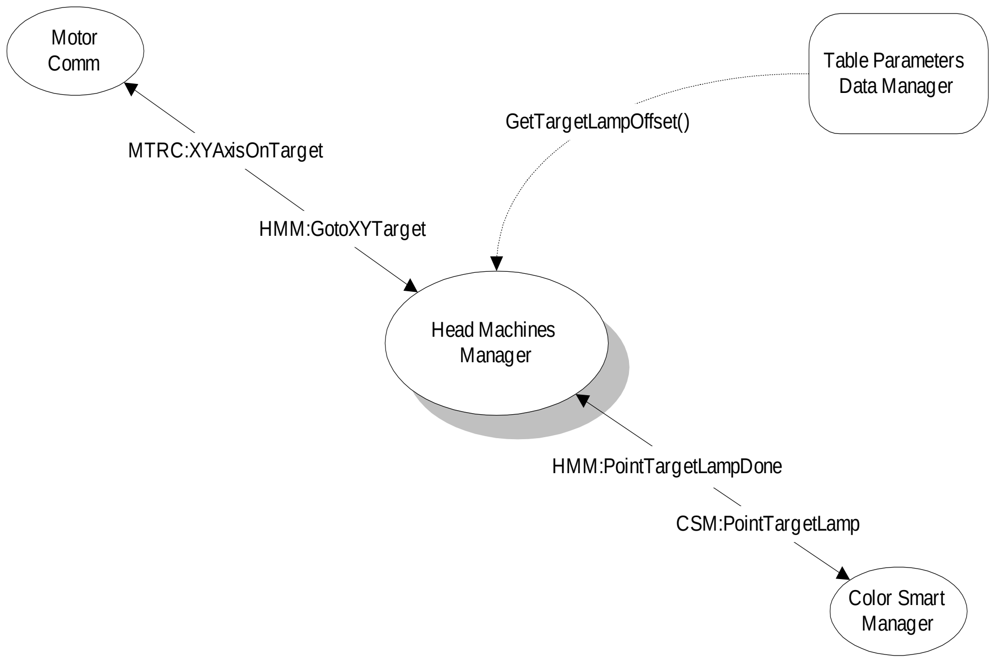


### 6.14.1.5. HMM System Interaction, Point Target Lamp

There are some instances during the Color Smart operation where the target lamp will be needed to be point to a specific location. This message accomplishes just that.

Since the controller knows the offset of the target lamp with respect to the measurement sensor, the NT Application can simply indicate where he wants the target lamp to be pointed.

CSM sends the “PointTargetLamp” request. The diagram shows all the machines involved in performing the process.




Note:

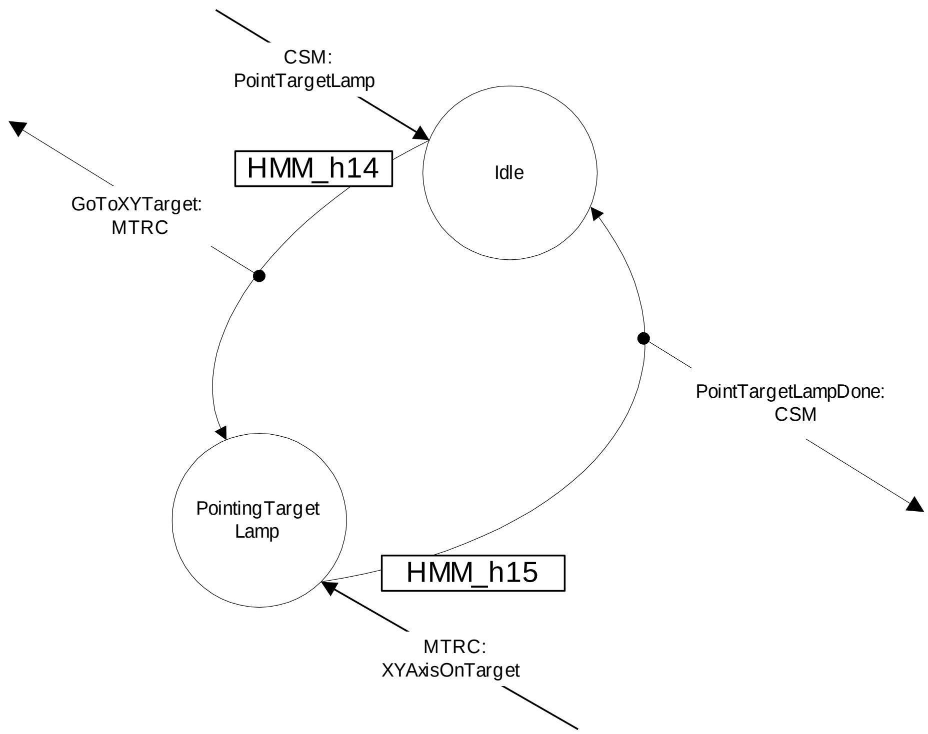
Activating and deactivating the target lamp requires sending another message.

### 6.14.1.5.1. HMM – PointTargetLamp, State Transitions

When the “Point Target Lamp” request is received, HMM simply takes the offset of the target lamp center with respect to the measurement sensor center, calculates where the measurement sensor should be in order for the target lamp to point to the requested position. CSM is notified once the request has been completed.

The “Point Target Lamp” message is accompanied by data.

Data1 = Target Lamp X Coordinate
Data2 = Target Lamp Y Coordinate




Note:

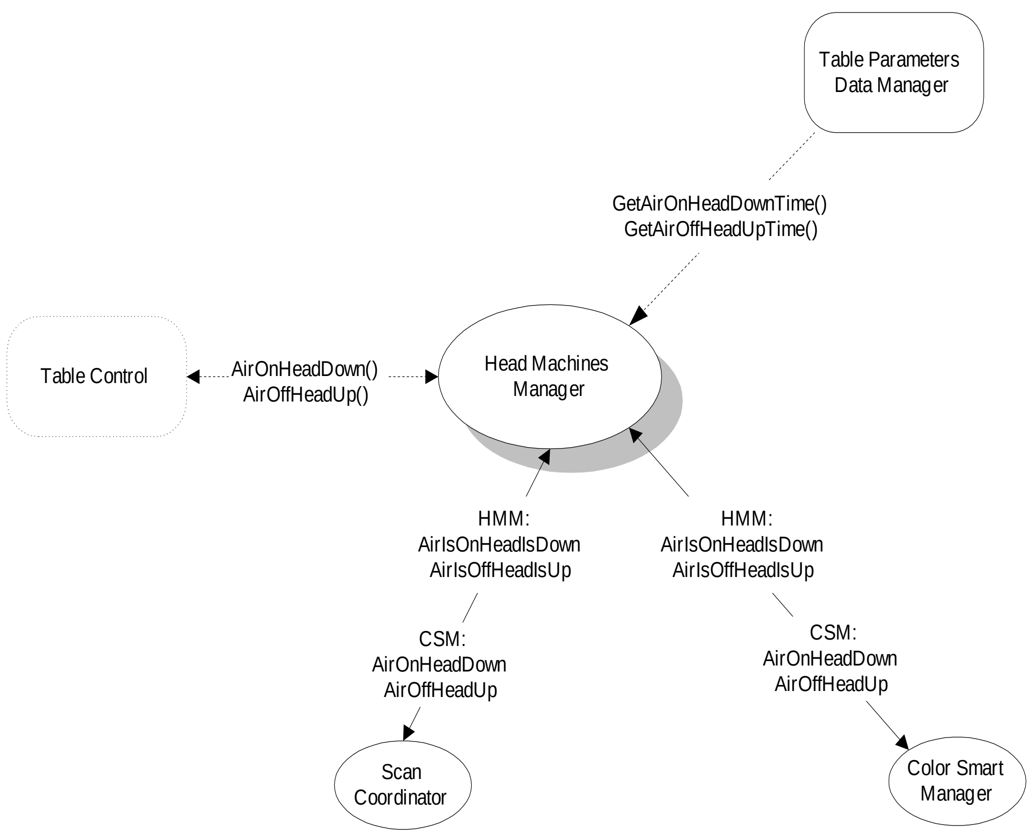
Unlike the “Move To Position” command this process does not have the settle delay. In addition to the “Turn Target Lamp On/Off”, this request is primarily used to show the end-user a specific point within the table or sheet. Consequently, there is no need to enable the Backlash Compensation when performing this process.

### 6.14.1.6. HMM – System Interaction, AirOnHeadProbeDown / AirOffHeadProbeUp

This is a direct command from either CSM or HOC telling him to activate or deactivate the table air. The diagram shows all the machines involved in performing the process.

Note:

The Probe Head automatically drops when the air is turned on. The Head automatically goes up when air is turned off.

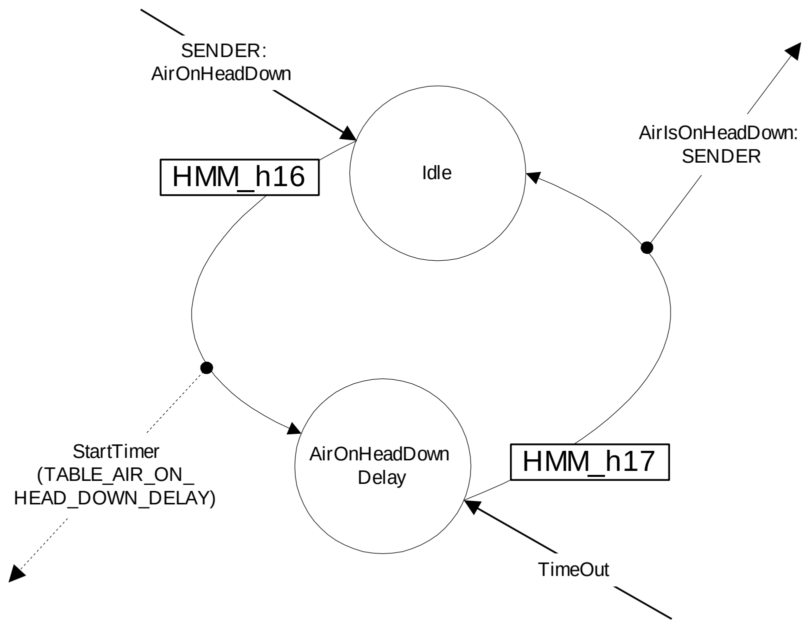


#### 6.14.1.6.1. HMM – Air On Head Down, State Transitions

CSM or HOC generates the "AirOnHeadDown" message.

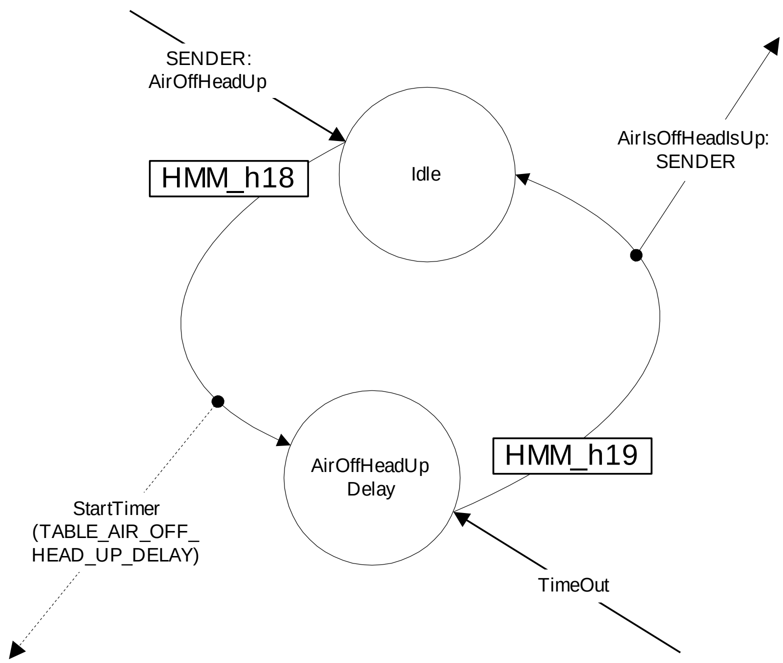
When the message is received by HMM, it activates or deactivates the air using the services of TC. It then performs a "Wait" to ensure that the head is either fully lowered or fully raised before he goes back to the idle state. Once HMM is in the idle state, a measurement command can immediately be processed and executed.

The SENDER is notified about the completion of the request.



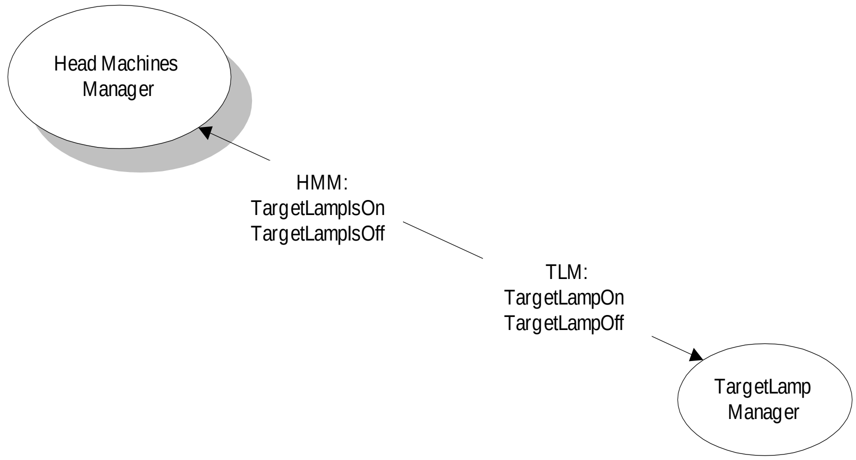
### 6.14.1.6.2. HMM – Air Off Head Up, State Transitions

CSM or HOC generates the “AirOffHeadUp” message.



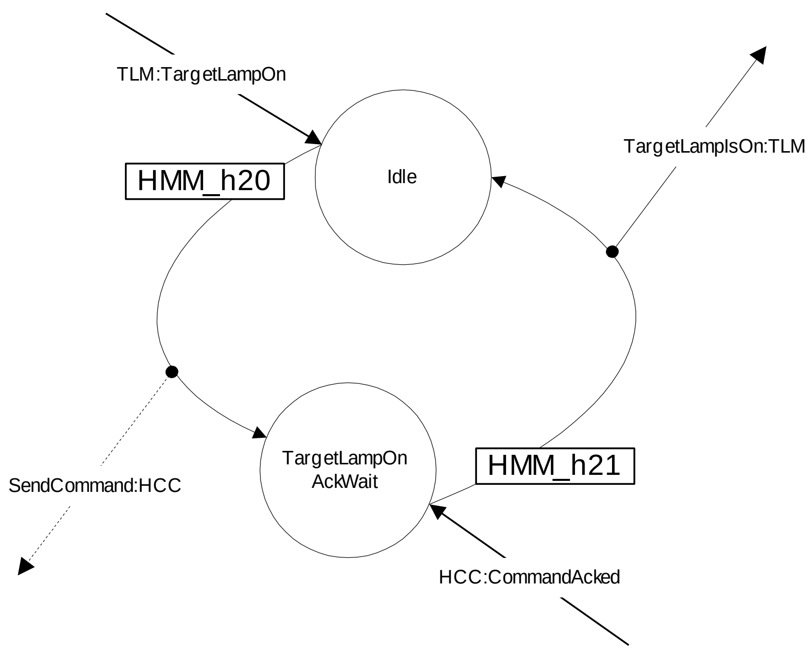
### 6.14.1.7. HMM – System Interaction, TargetLampOn / TargetLampOff

This is a direct command from TLM telling him to activate or deactivate the Target Lamp. The diagram shows all the machines involved in performing the process.



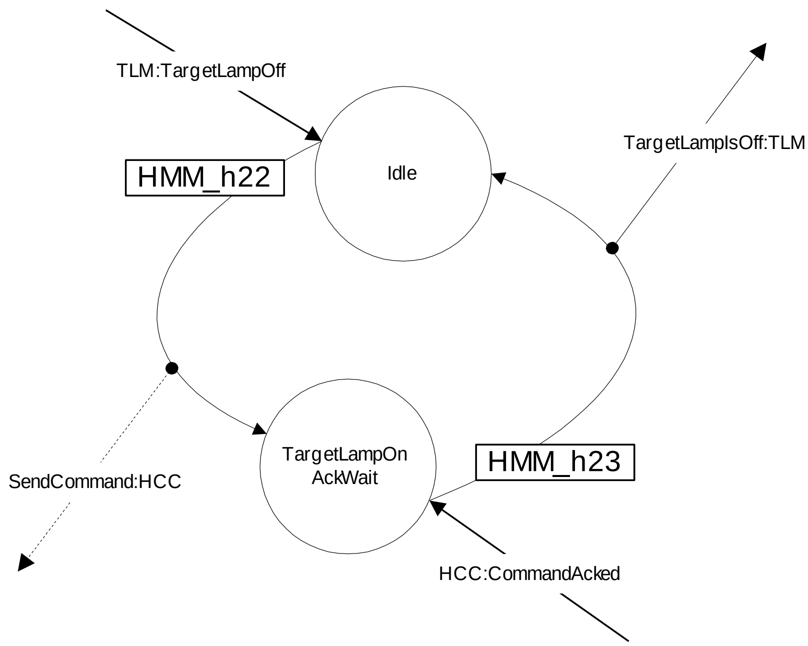

6.14.1.7.1. HMM – Target Lamp On, State Transitions

TLM generates the "TargetLampOn" message.



### 6.14.1.7.2. HMM – Target Lamp Off, State Transitions

TLM generates the "TargetLampOff" message.

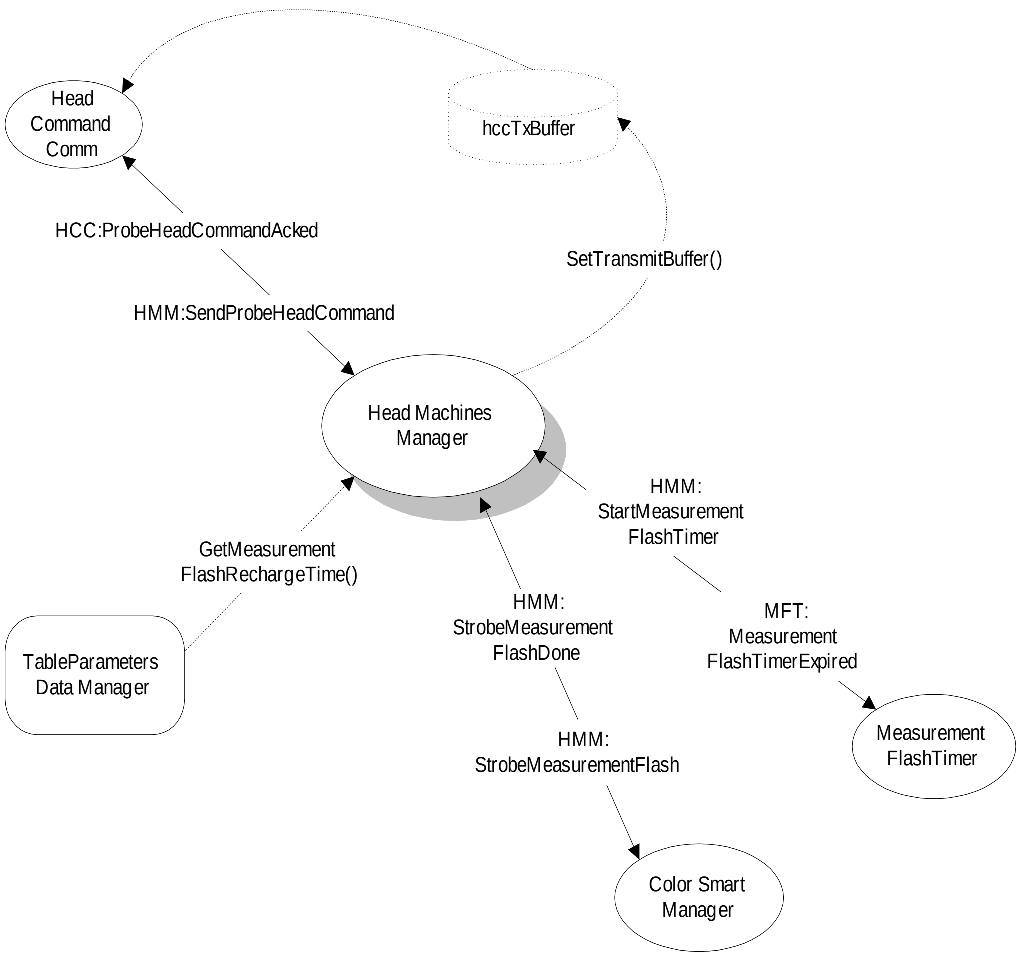


### 6.14.1.8. HMM – System Interaction, StrobeMeasurementFlash

This is a direct command from either CSM him to generate a strobe of the measurement lamp. The probe head module itself generally controls the measurement flash when a Measurement Command is processed. The StrobeMeasurementFlash command allows us to fire the flash at will.

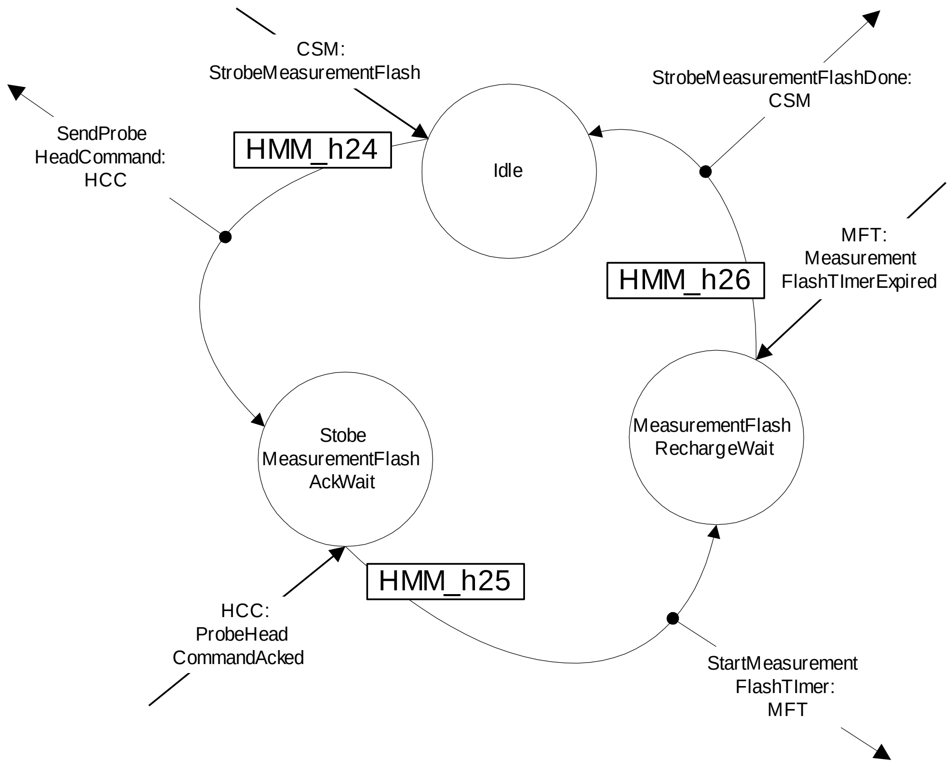
NOTE: This operation is only permitted by CSM when he is in the “SERVICE mode”.

The diagram shows all the machines involved in performing the process.



### 6.14.1.8.1. HMM – Strobe Measurement Flash, State Transitions

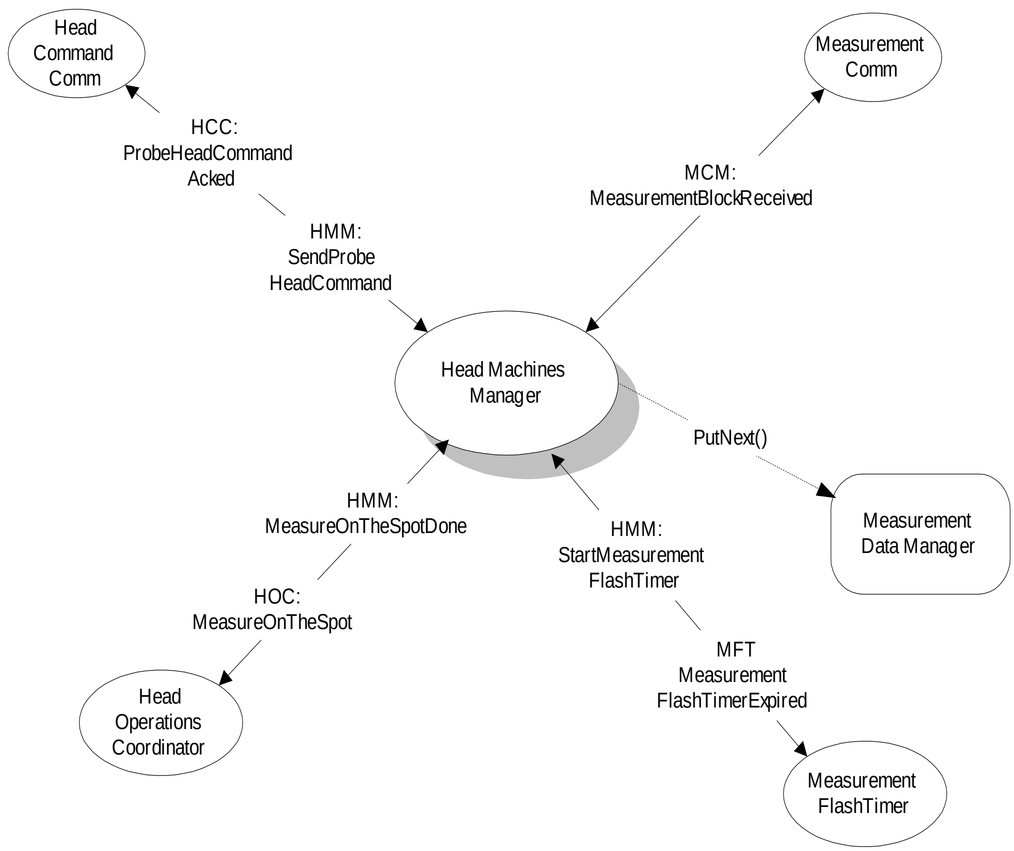
When the message is received by HMM, it fills up the transmit buffer with the probe head command and requests HCC to begin a transmission. HMM is informed once HCC receives the reply (ACK). HMM then uses MFT and performs a “Wait” to ensure that the flash has been given enough time to recharge before he goes back to the idle state. This is to ensure that HMM is ready to process and execute another “Strobe Measurement Flash” command when it is in the idle state.



### 6.14.1.9. HMM – System Interaction, MeasureOnTheSpot

"Measure on the Spot" is a direct command from HOC telling him to perform a point measurement without moving the head. Again, HOC ensures that the probe head is properly raised or lowered. This process uses the services of the Measurement Data Manager to store the measurement data.

The diagram shows all the machines involved in performing the process.




**IMPORTANT: This procedure maybe used in conjunction with the Track Ball Moves where the Backlash Compensation is disabled. It is up to the NT Application to keep track of the backlash and make appropriate adjustments in this case to determine the "TRUE" position of the measurement sensor.**

**The MeasureAtPoint request is used to accurately measure a specific location.** **See page 152 for more information.**

**There is an assumption in this behavior:**

**MeasurementFlashRechargeTime (approx 20 msec)**
**(is Greater Than) the**
**MeasurementDataTransferTime (approx 6 msec)**

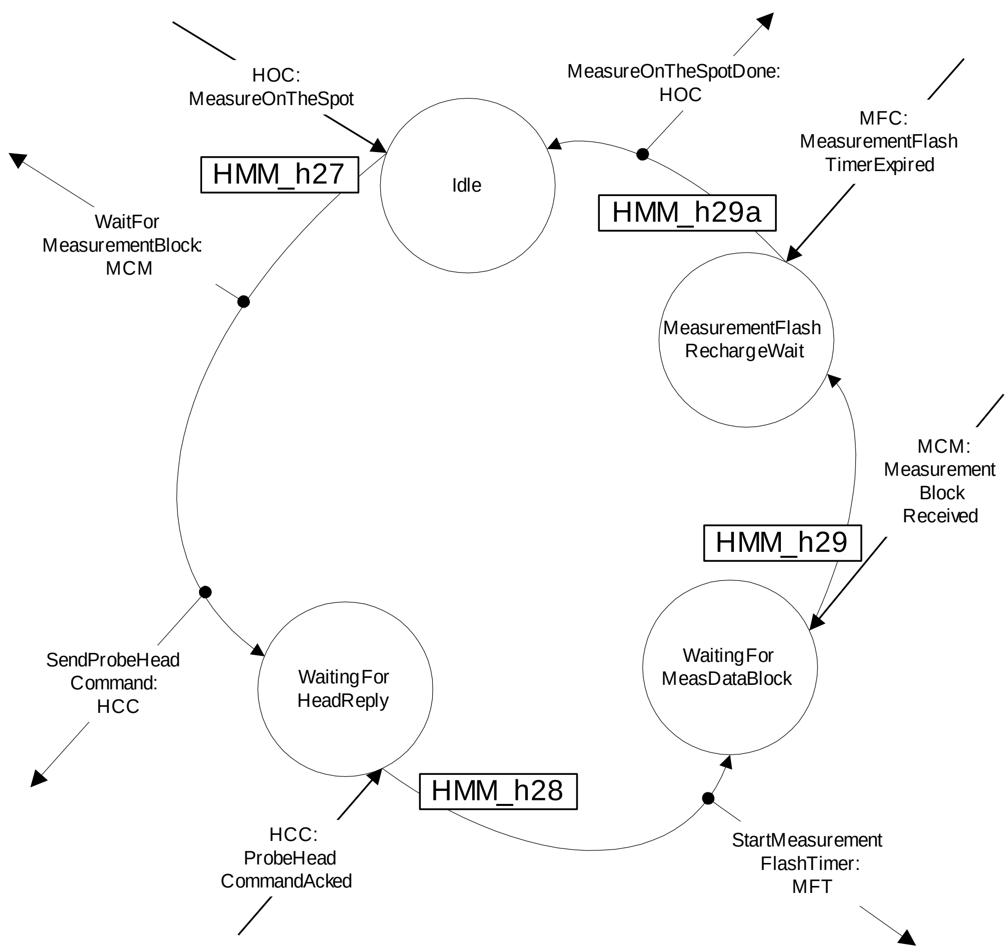
### 6.14.1.9.1. HMM – Measure On the Spot, State Transitions

The "MeasureOnTheSpot" message is generated by HOC.

When the message is received by HMM, it fills up the transmit buffer with the probe head "MEASUREMENT" command. A request is made to HCC to begin a transmission. HMM is informed once HCC receives the reply (ACK). HMM then waits for MCM to signal him that the measurement data has been fully received. HOC is informed about the completion of the request.

**IMPORTANT: The DMA Peripheral is activated prior to changing the machine's state from IDLE to "WAITING FOR HEAD REPLY". This is to ensure that the DMA for Serial Reception is ready by the time the probe head initiates the Measurement sequence. See "Synchronous Comm IRQ (ISCM) Task Description" on page 64 for more detail.**

**CSM is responsible for "Resetting" MDM, before making this request. This procedure simply inserts the next measurement into the next available slot of MDM.**

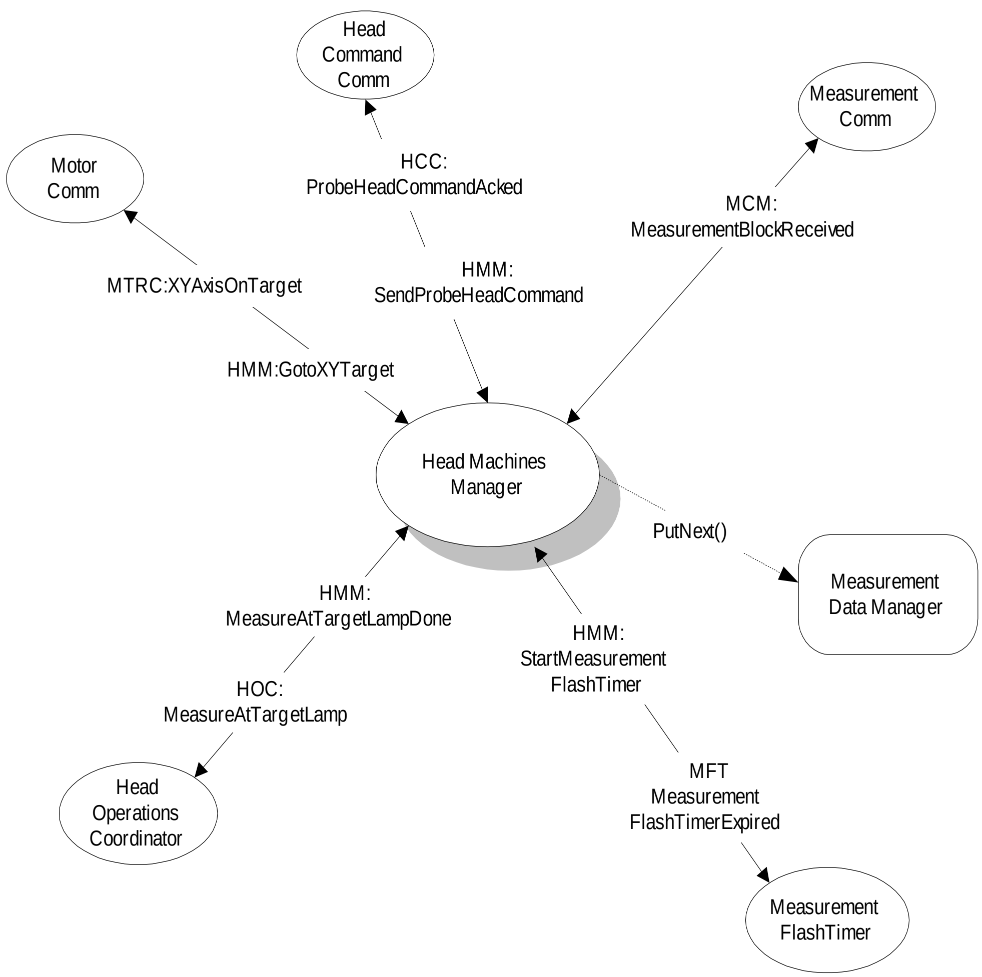


### 6.14.1.10. HMM – System Interaction, MeasureAtTargetLamp

MeasureAtTargetLamp is a command from either HOC, telling him to perform a point measurement where the target lamp is currently pointing. In this scenario, the end-user would normally be using the trackball to select the point to be measured using the Target Lamp.

**IMPORTANT: Measuring the "Target Lamp Point" requires precise positioning. Therefore, the backlash compensation mechanism of MTRC will be enabled.**

The diagram shows all the machines involved in performing the process.

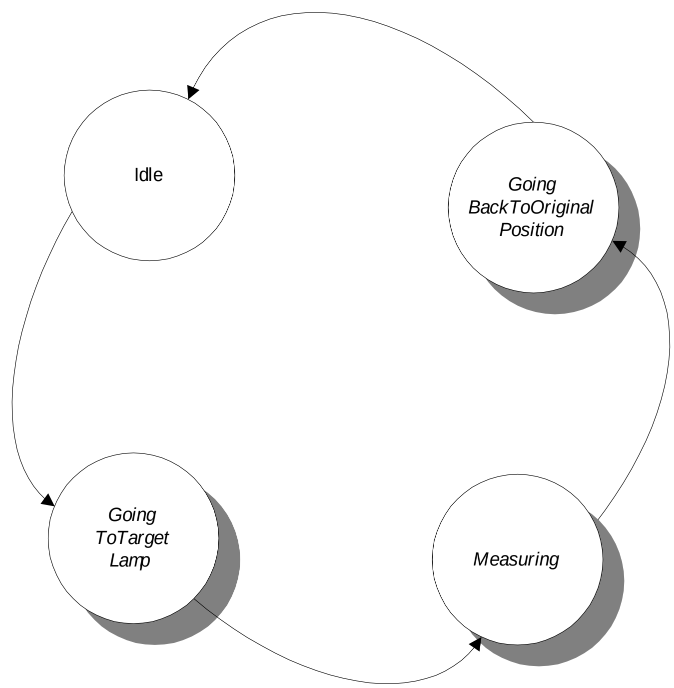


#### 6.14.1.10.1. HMM – Measure At Target Lamp, State Transitions Level 1

HOC generates the "MeasureAtTargetLamp" message.

This request causes the probe head measurement sensor to go to the Target Lamp's center point, make a measurement and then go back to its original position. This implies that the Target Lamp would again be pointing to the point that was just measured by the time HMM goes back to idle. HOC is informed about the completion of the request.

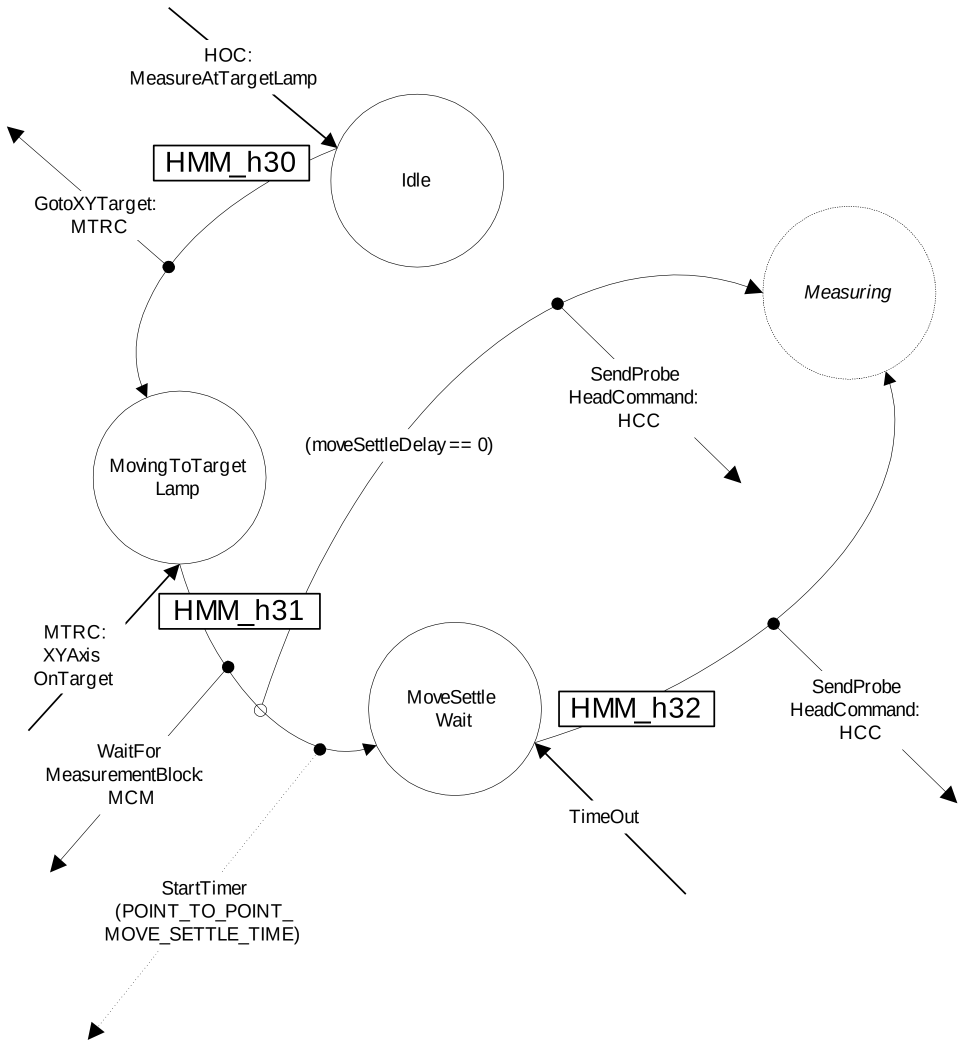
The diagram below shows this behavior.




**IMPORTANT: CSM is responsible for "Resetting" MDM, before making this request. This procedure simply inserts the next measurement into the next available slot of MDM.**

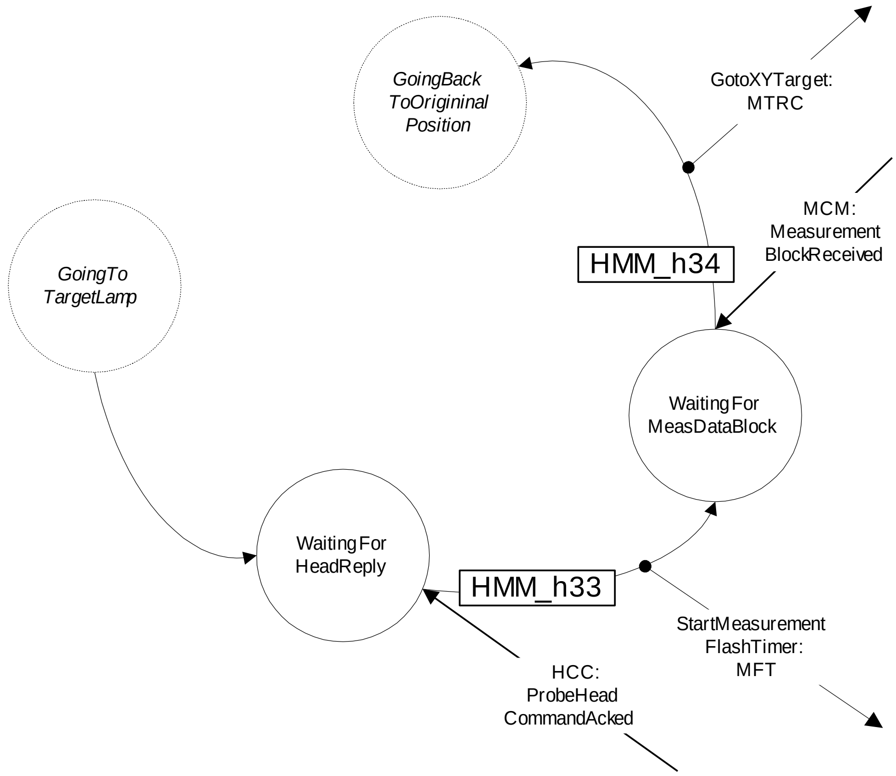
### 6.14.1.10.2. HMM – Measure At Target Lamp, State Transitions Level 2

This is an expanded view of the “Going To Target Lamp” behavior.



6.14.1.10.3. HMM – Measure At Target Lamp, State Transitions Level 2, continued . . .

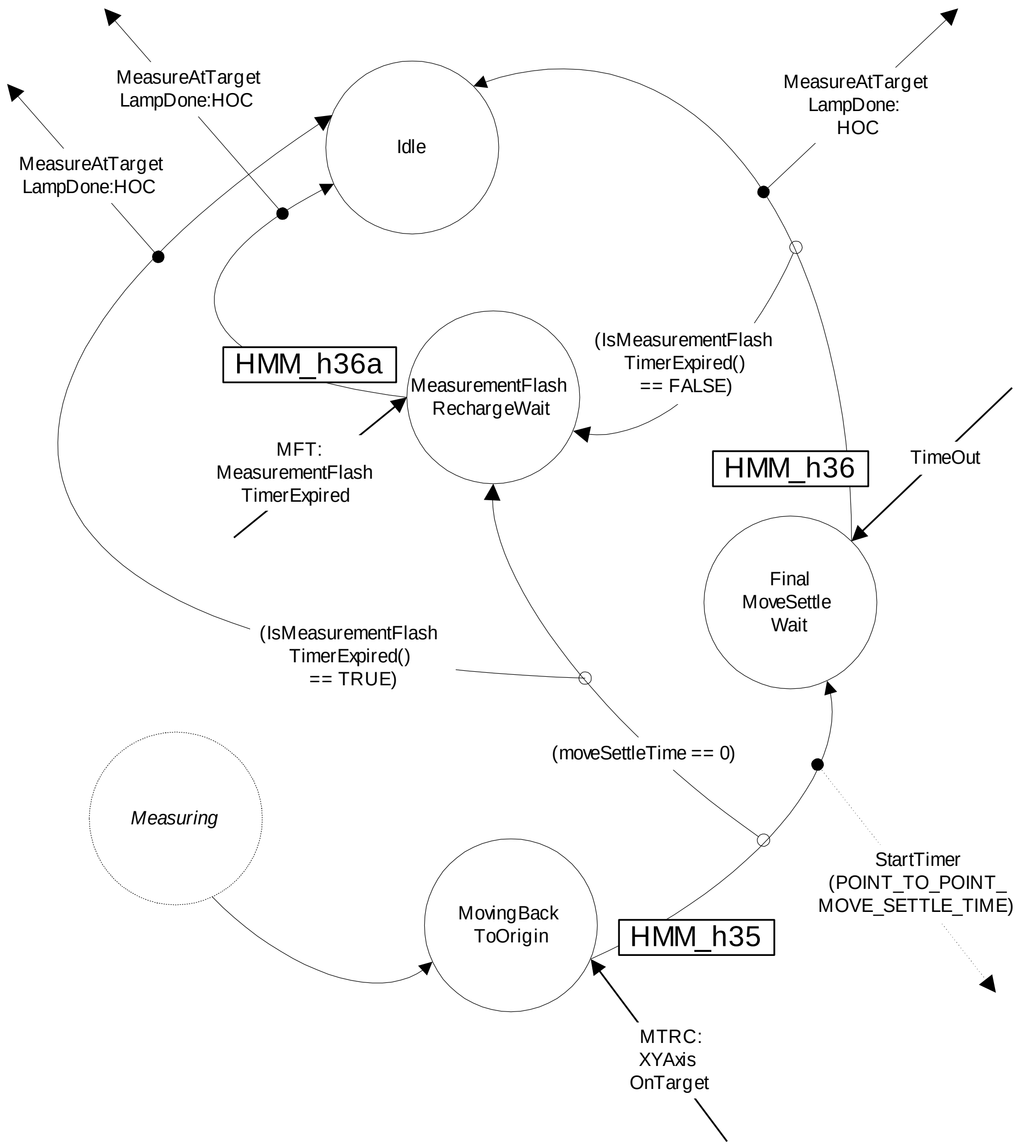
This is an expanded view of the "Measuring" behavior.



### 6.14.1.10.4. HMM – Measure At Target Lamp, State Transitions Level 2, continued . . .

This is an expanded view of the "Going Back To Original Position" behavior.

Note: The Measurement Flash Timer check as well as the Move Settle delay in this behavior allows a "Measure On The Spot" request to be executed immediately (if needed).



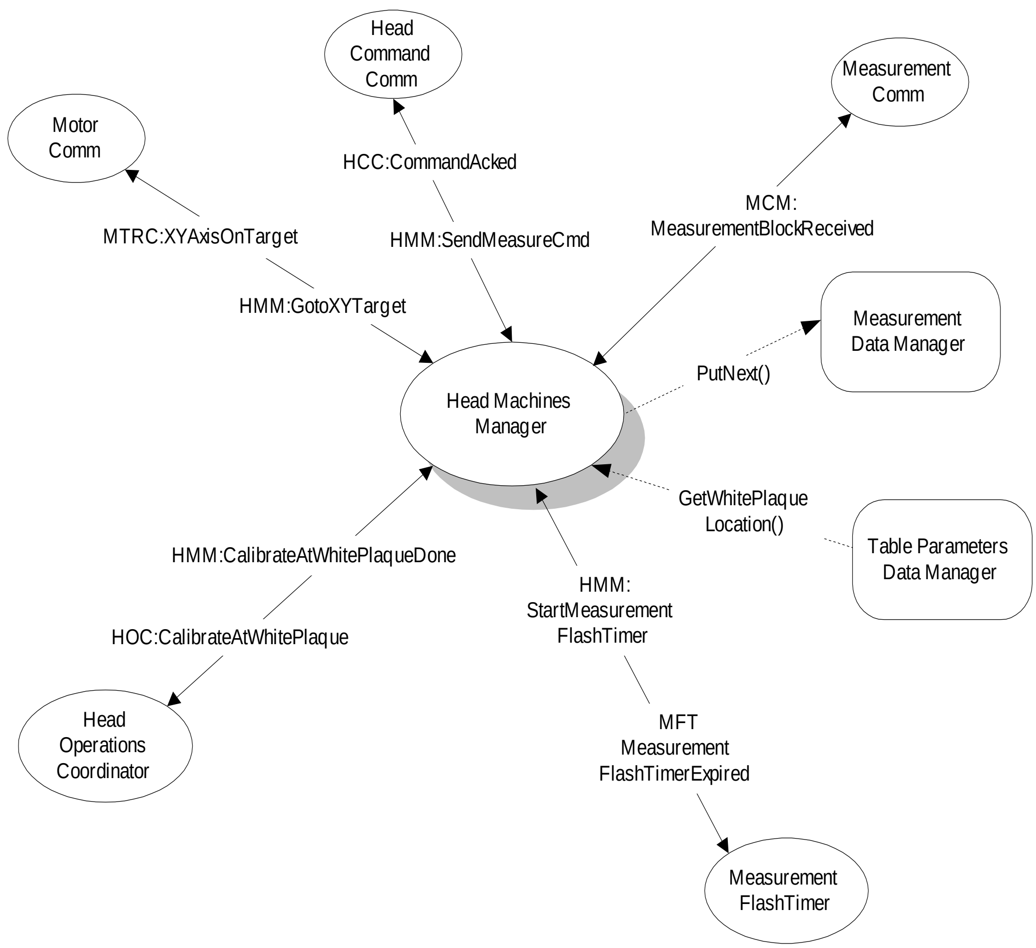
### 6.14.1.11. HMM – System Interaction, CalibrateAtWhitePlaque

The diagram shows all the machines involved in performing the process.

Note: HOC generates the “CalibrateAtWhitePlaque” message. As stated earlier, HOC is responsible making sure that the probe head is correctly set (Raised or Lowered), before sending this request.

A white plaque calibration without the use of the flash is supported by HMM. This message is accompanied by data:

Data 1 = 0 - measure Without Flash
1 – measure With Flash



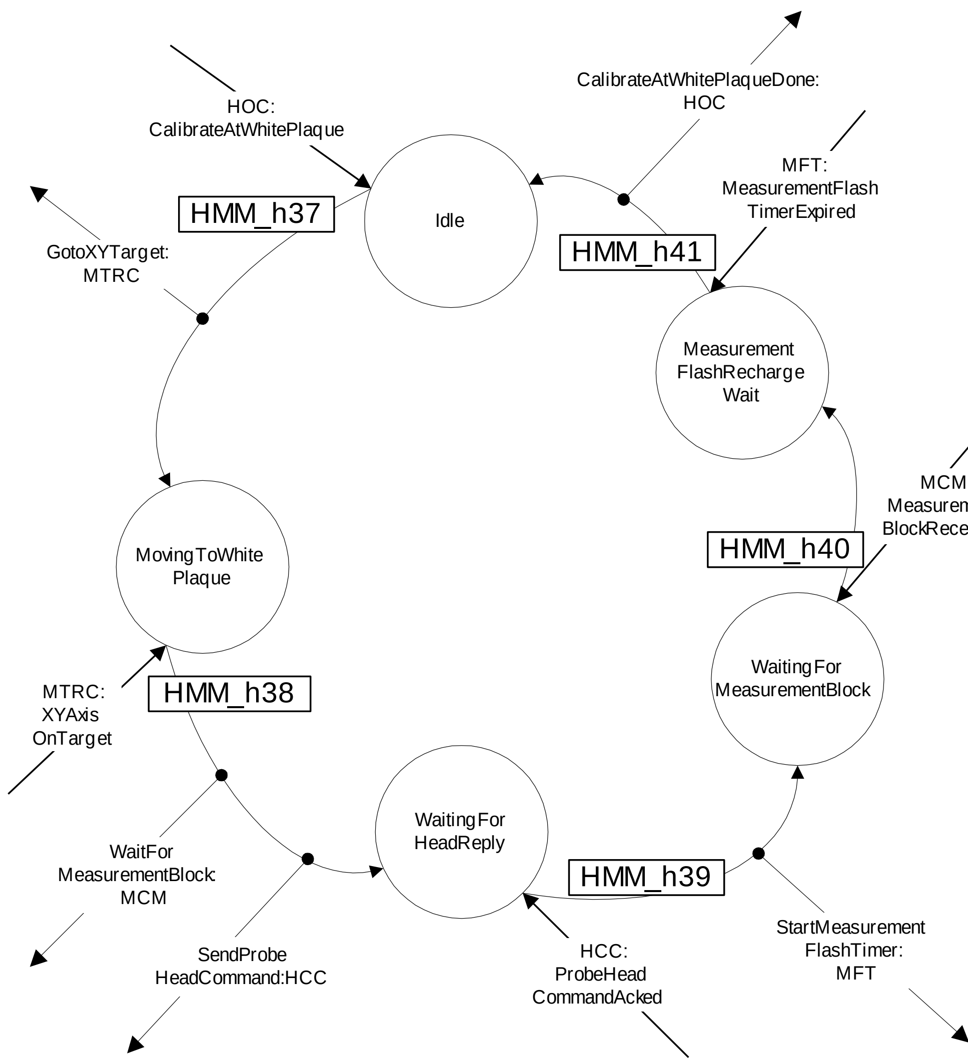

**IMPORTANT: CSM is responsible for “Resetting” MDM, before making this request. This procedure simply inserts the next measurement into the next available slot of MDM.**

### 6.14.1.11.1. HMM – Calibrate At White Plaque, State Transitions

HMM knows where the White Plaque is located within the table. Once the "CalibrateAtWhitePlaque" message is received, HMM sends a request to MTRC to go to the white plaque position. When the probe head settles, a measurement sequence is initiated. HOC is informed about the completion of the request before going back to the idle state.

NOTE: Since the White Plaque reference area is fairly large, there is no Move Settle Wait after move is complete. The probe head will be left at the White Plaque after this process is completed.

Consecutive "Calibrate at White Plaque" request will cause MTRC not to move and instead immediately replies with an "XYAxisOnTarget" message.

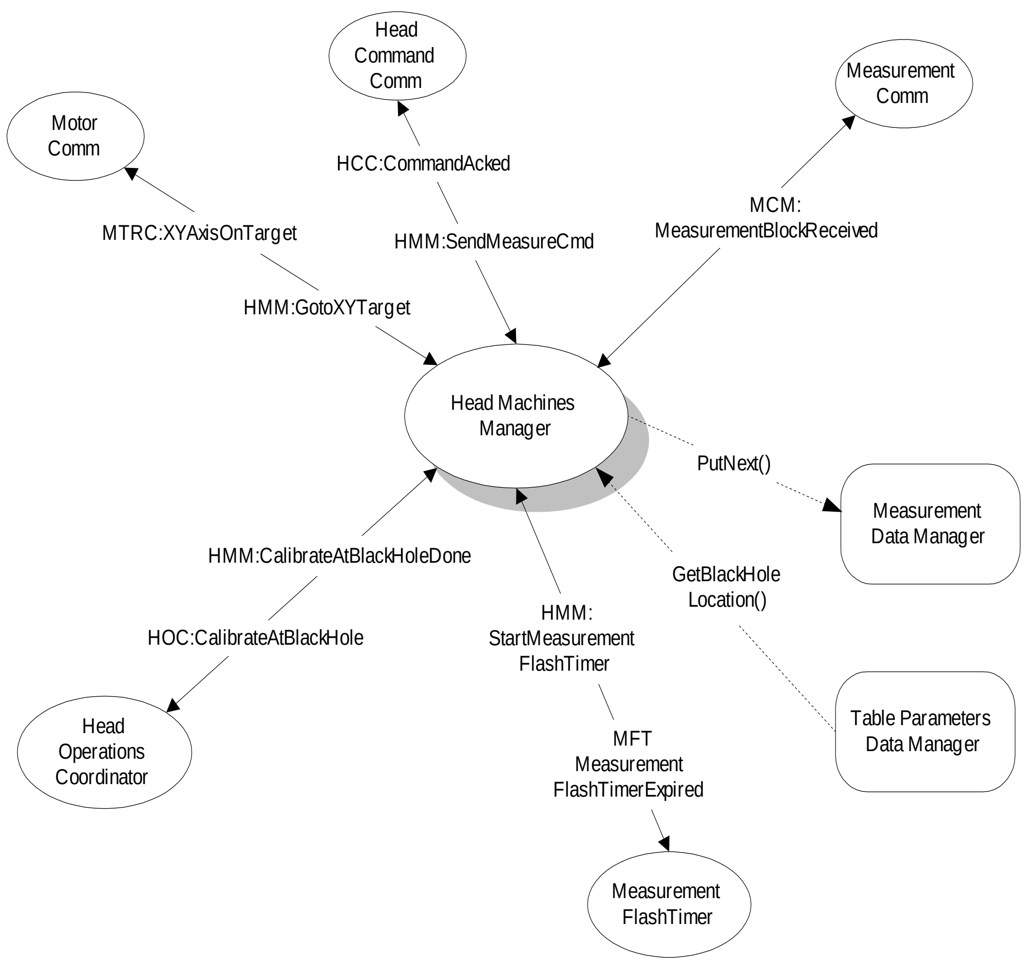


### 6.14.1.1. HMM – System Interaction, CalibrateAtBlackHole

HMM knows where the Black Hole is located within the table. This process is exactly the same as that of the White Plaque calibration except for the location of the measurement.

The Black Hole calibration is generally done before the White Plaque calibration.

The diagram shows all the machines involved in performing the process.



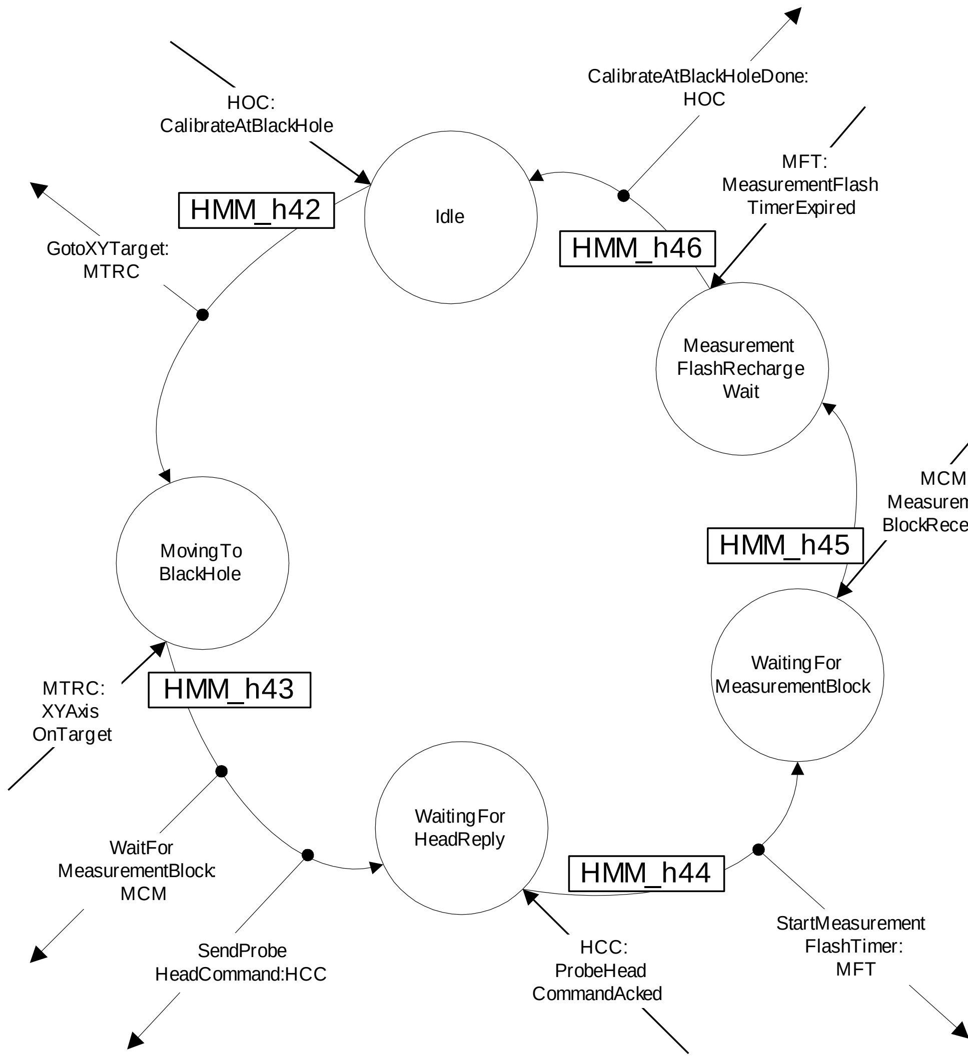

**IMPORTANT: CSM is responsible for "Resetting" MDM, before making this request. This procedure simply inserts the next measurement into the next available slot of MDM.**

## 6.14.1.1.1. HMM – Calibrate At Black Hole, State Transitions

HOC generates the “CalibrateAtBlackHole” message.

NOTE: Since the White Plaque reference area is fairly large, there is no Move Settle Wait after move is complete. The probe head will be left at the White Plaque after this process is completed.

Consecutive “Calibrate at White Plaque” request will cause MTRC not to move and instead immediately replies with an “XYAxisOnTarget” message.



### 6.14.2. HMM – Alignment Scans

There are 2 types of Alignment Scans:

a) Target Alignment Scan
b) Alignment Scan

#### 6.14.2.1. Target Alignment Scan

Is used during a Color Smart diagnostics session where the Probe Head is calibrated to make sure that the measurement sensor is precisely on target. The moves generated by the Target Alignment process is similar to a point to point move in that the Firmware sends the Motor to a given direction, one step at a time while measuring at the end of every step.

**IMPORTANT: The NT Application is expected to position the head at the initial point with the backlash already compensated at the time the Target Alignment request is made. The "second, reverse" pass of the Target Alignment procedure should leave the backlash uncompensated. This is used to determine the maximum backlash of the table.**

**The step-by-step behavior of the Target Alignment Scan ensures the positional accuracy of the measurements.**

**Below is a more detailed description of how the whole Target Alignment process works, from the Color Smart system's level point of view:**

The purpose of this procedure is to determine the precise location of the measurement aperture with respect to the projected target the customer uses to point to locations on the table. The nominal X and Y distance between the measurement aperture and projected target is known. For the test, the operator points to a dot on a special test target, then the computer locates the center of the dot using the measurement aperture to determine the actual distance between the two.

The completion of this procedure updates TargetLampOffset variable the to it's true value. The controller board itself uses this variable. Another variable, which gets updated, is the MaximumTableBacklash, which is stored by the NT Application. The NT Application uses this to determine how the current backlash values for each axis when the trackball moves are being performed.

The Target Alignment Procedure goes as follows:

1. The operator carefully points to the black dot on the QA 2180 alignment target. There are cross hairs to help. The dot must be the same size or slightly smaller than the aperture.
2. The computer moves and measures at a point 2cm above and 2cm to the left of the dot to determine a "white" signal.

NOTE: All signals are measured with visual response. For Color Smart, we use visual reflectance (not density) for this test.

3. The computer moves with an anti-backlash move to a measurement point 25 steps to the left of the dot using the nominal_aperture_target_distance(XY) as the distance between the measurement aperture and the projected target.
4. The computer scans 50 steps across the target making one measurement each step. The location of the lowest signal (or reflectance) is the location of the center of the dot.
5. The NT Application makes coordinated requests to the controller card. The card initiates the scans, 50 steps from right to left (across the dot) with a measurement in each step. The location of the center of the dot determined by this scan, will be different than that for the left to right scan by the distance equivalent to the Maximum Table Backlash of the X axis. This is how we determine and store the Maximum Table Backlash.
6. The Y axis is scanned in a similar manner to determine the aperture-to-target distance (TargetLampOffset) as well as the maximum backlash for the Y axis.

Other Notes:

For ColorSmart, each move, the white reading, and each of the 4 scans should be a separate NT command. The NT Application will find the center of the dot for each scan by calculating visual reflectance for each measurement and searching for the location with minimum reflectance. The purpose of measuring white in Autosmart was to set an initial minimum for the search and may not be necessary for Color Smart. Since Color Smart is working with reflectance instead of signals, we can use a nominal value of 80 percent reflectance for white if we need one. We would expect the minimum reflectance at the center of the dot to be below 40 percent.

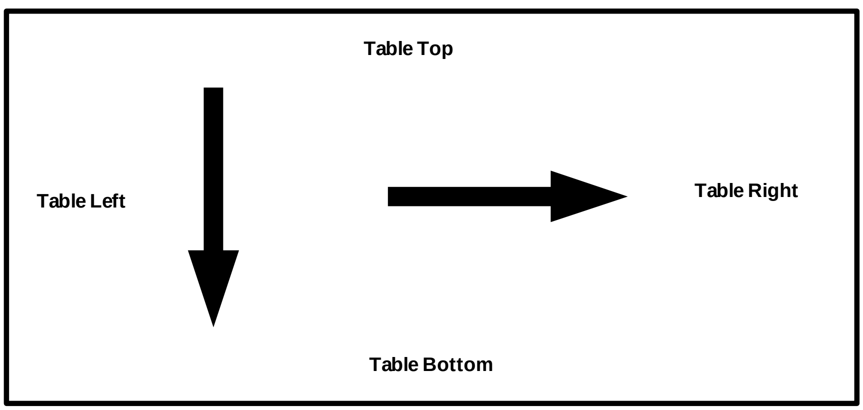
### 6.14.2.2. Alignment Scan

Is mainly used to recalibrate the coordinate of the measurement points based on the placement of the sheet on the table. The exact placement of the sheet is determined by scanning an alignment bar or a set of alignment bar (points). Unlike the Target Alignment, the move performed is smooth and continuous, very much like a color bar move. However, the implementation is slightly different since the scan can be performed in 2 directions unlike the color bar scan.

**IMPORTANT: The "DoAlignmentScan" request is used for both the "Auto Alignment Scan" and the "Manual Alignment Scan" processes of the Color Smart.**

To prevent backlash errors, the Alignment Scan will only be performed in 2 directions as shown below. CSM as the initiator of the request, is responsible for making sure the Target Alignment data follows this rule.

These 2 directions are also known as the **"Anti-backlash"** directions for the reason that they do not induce any backlash to the moves.



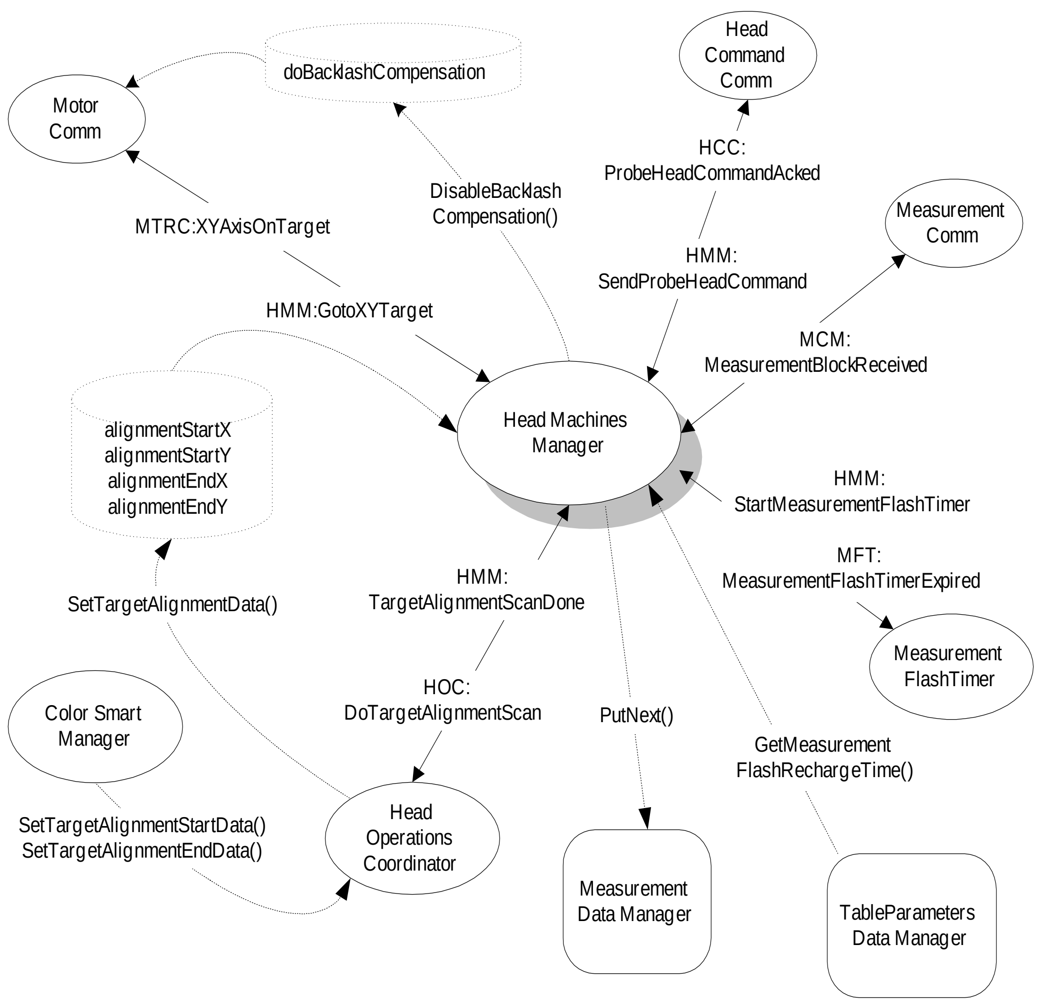

See "HMM – System Interaction, DoColorBarScan on page 149 to get more information on how this process is performed.

### 6.14.2.3. HMM – System Interaction, DoTargetAlignmentScan

**IMPORTANT: In order to provide the highest resolution when searching for TARGET Alignment Point, the motors make 1 step at a time in a given direction. This process of searching also requires precise positioning.**

**CSM is responsible for "Resetting" MDM, before making this request. This procedure simply inserts the next measurement into the next available slot of MDM.**

CSM sets the alignment coordinate data of HOC before he issues the request to HOC. HOC in then makes a request to HMM. The diagram below shows all the machines involved in performing the process.



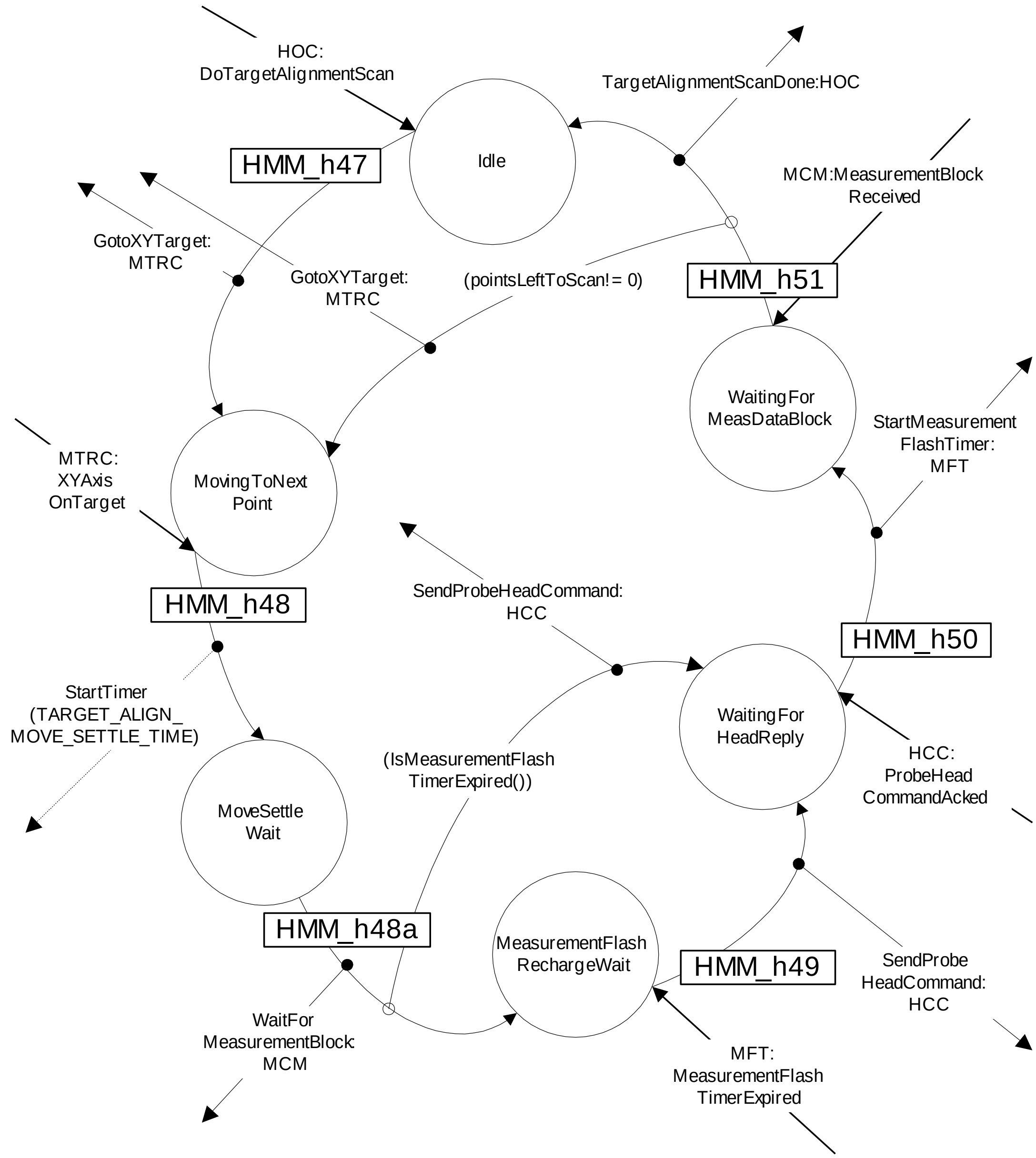

NOTE: See CSM – (Scanning Mode Idle), Do Alignment Scan, State Transitions on page 40 for more detail about the alignment scan feature.

### 6.14.2.3.1. HMM – DoTargetAlignmentScan, State Transitions

The “Do Target Alignment Scan” message is generated by HOC.

Based on the starting and the ending coordinates, HMM determines the direction to which the probe head should be moving. The machine then generates the message requests to MTRC to perform the moves one step at a time. HMM measures as it moves along the way.

Note: For accuracy, a “Move Settle Wait” needs to be performed before the measurement. The first point measurement assumes that the Measurement Flash is ready.


HOC:
DoTargetAlignmentScan
TargetAlignmentScanDone:HOC
HMM_h47
Idle
MCM:MeasurementBlock
Received
GotoXYTarget:
MTRC
GotoXYTarget:
MTRC
(pointsLeftToScan!= 0)
HMM_h51
WaitingFor
MeasDataBlock
StartMeasurement
FlashTimer:
MFT
MTRC:
XYAxis
OnTarget
MovingToNext
Point
SendProbeHeadCommand:
HCC
HMM_h48
HMM_h50
StartTimer
(TARGET_ALIGN_
MOVE_SETTLE_TIME)
WaitingFor
HeadReply
(IsMeasurementFlash
TimerExpired())
HCC:
ProbeHead
CommandAcked
MoveSettle
Wait
HMM_h48a
MeasurementFlash
RechargeWait
HMM_h49
SendProbe
HeadCommand:
HCC
WaitFor
MeasurementBlock:
MCM
MFT:
MeasurementFlash
TimerExpired

### 6.14.2.4. HMM – System Interaction, DoAlignmentScan

Just like the color bar scan, the Start Offset is used to make sure that the probe head is already moving at a constant speed before the first point target measurement is taken. The End Offset on the other hand is used to make sure that the last measurement has already been taken before the deceleration begins.

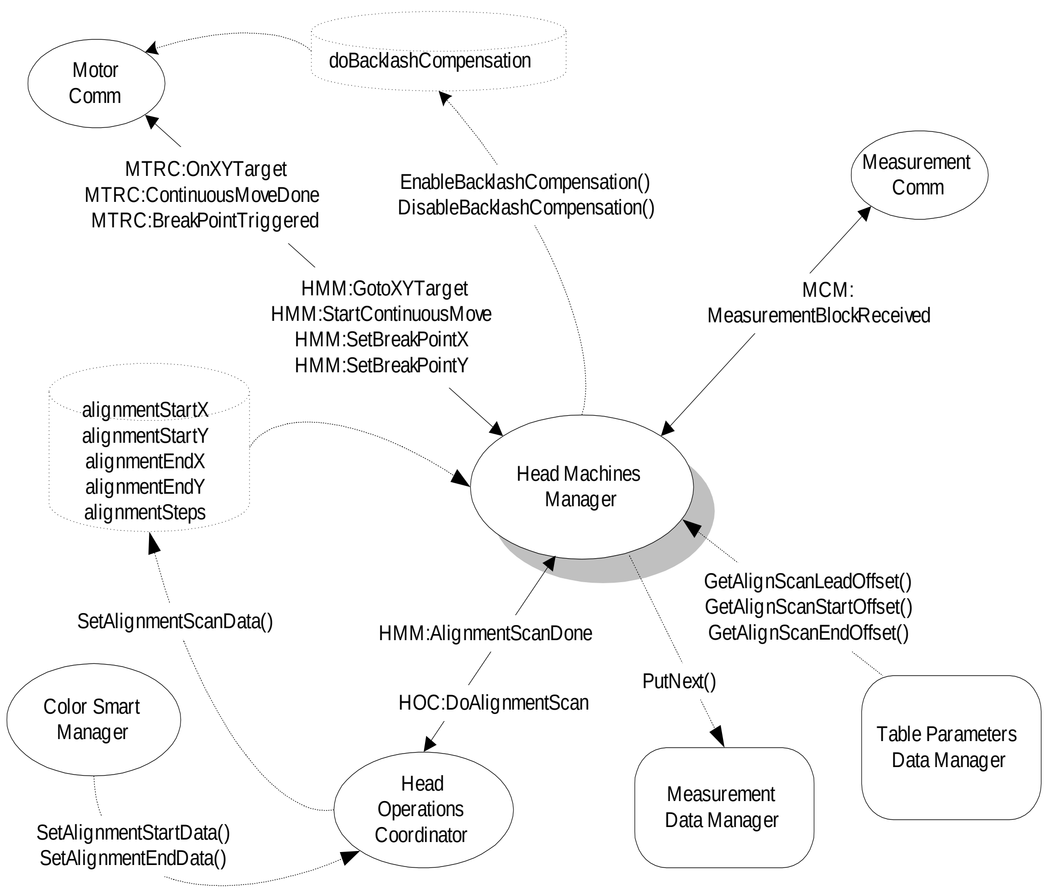
The Lead Offset is a special parameter used to compensate for all System latencies that will be involved in the Alignment Scan process. Since this move profile used is relatively slow, and the measurement step intervals are short, the Lead Offset is typically set to 0.

**IMPORTANT:**

**This process assumes that the move is slow enough for the Measurement Flash to be fully recharged before the next breakpoint occurs.**

**It is possible to miss the "End Point" if the "Alignment Steps" specified does not evenly divide the "Start to End Point Ranges"**

The diagram shows all the machines involved in performing the process.

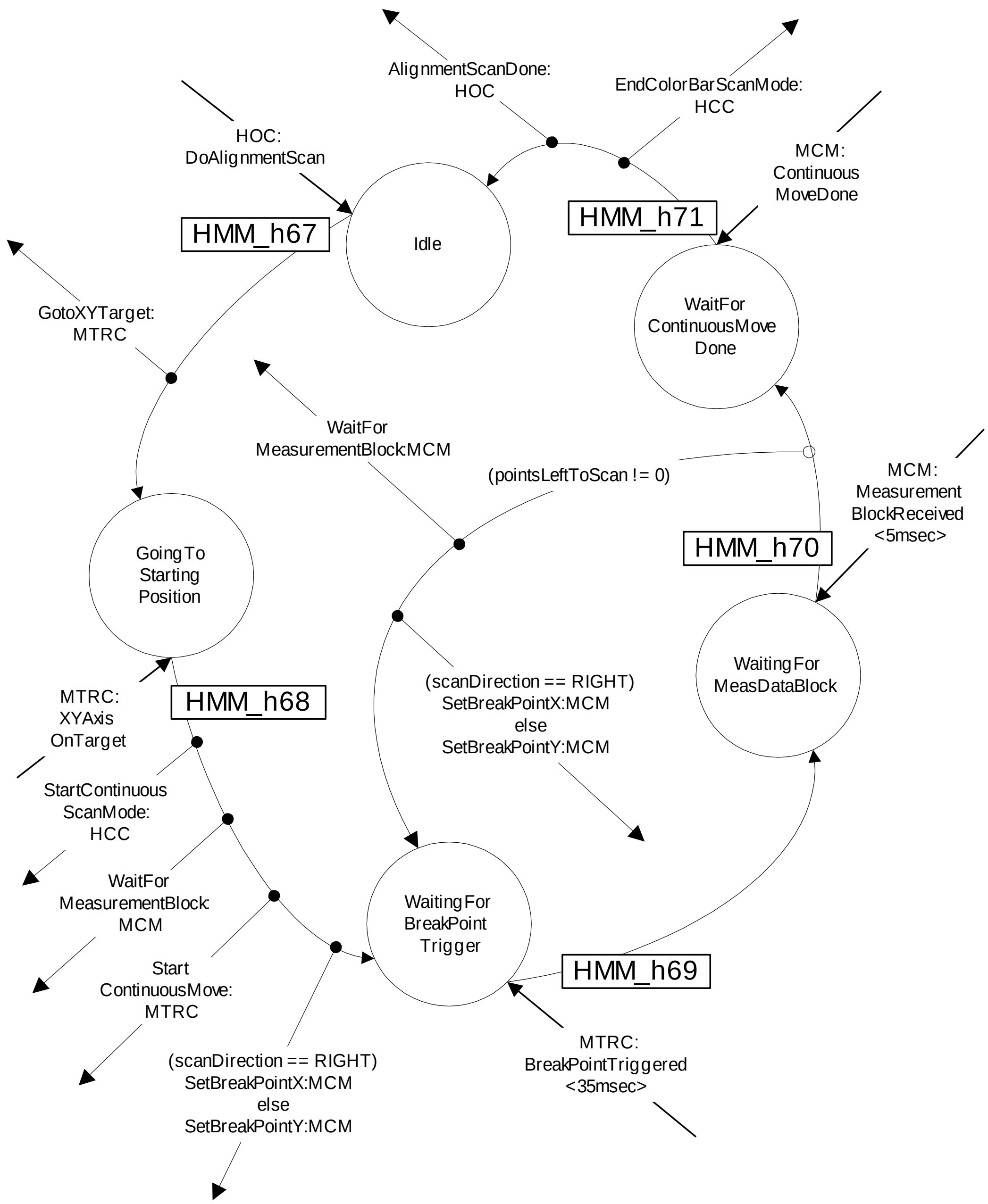


### 6.14.2.4.1. HMM – Do Alignment Scan, State Transitions Level 1

The “DoAlignmentScan” message is generated by HOC. The state transitions are very similar to that of the color bar scan.

**IMPORTANT:**

**The motor chipset’s break points are updated each time the measurement data block is received. Either breakpoint X or Y is updated based on the direction of the alignment scan.**



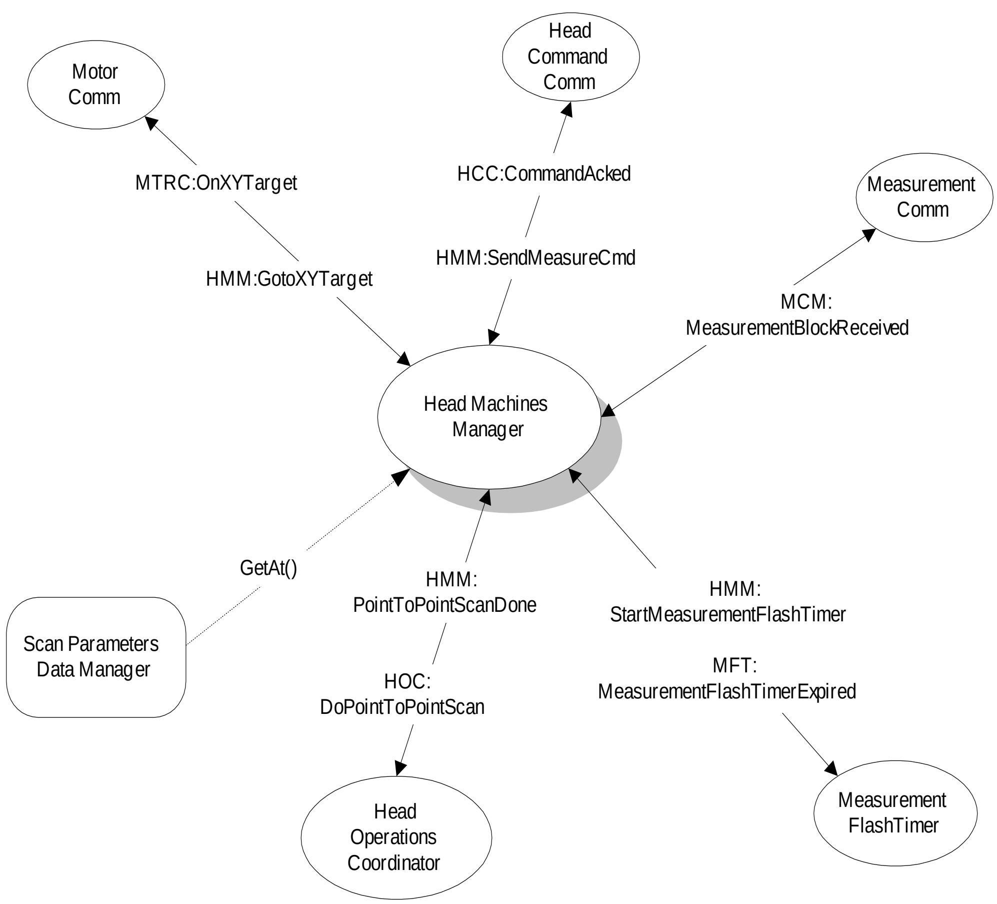
### 6.14.2.5. HMM – System Interaction, DoPointToPointScan

**IMPORTANT: The "Point to Point" measurement process requires precise positioning where the backlash should be compensated.**

**But instead of enabling the backlash compensation for all the point scan, we follow this rule: If the next move is going towards the "Anti-backlash" direction then backlash compensation will not be necessary. On the other hand, if the next move will introduce backlash, then the backlash compensation will be enabled for the next move. This significantly reduces the need for Color Smart to perform backlash to only about half of the Point to Point scans. This consequently reduces the total scan time.**

**Finally there is an assumption that the (Flash Recharge Time + Measurement Transfer Time (approx. 6 msec)) is greater than the step motor's Reverse settle time. See pages 162 and 167 for more information.**

The diagram shows all the machines involved in performing the process.



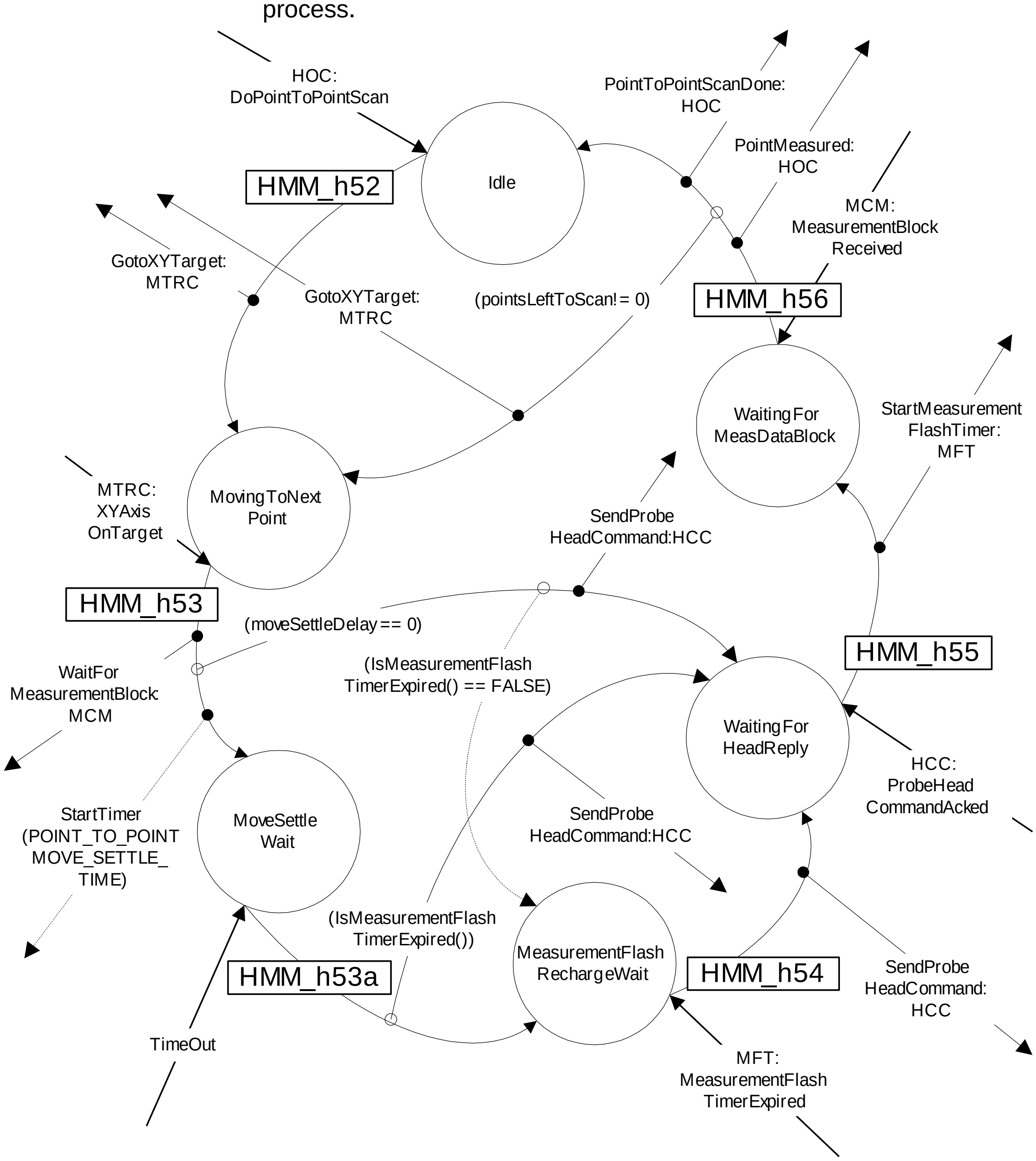
### 6.14.2.5.1. HMM – Point To Point Scan, State Transitions

The “DoPointToPointScan” message is generated by HOC.

Data1 = scan Start Point Index
Data2 = scan End Point Index

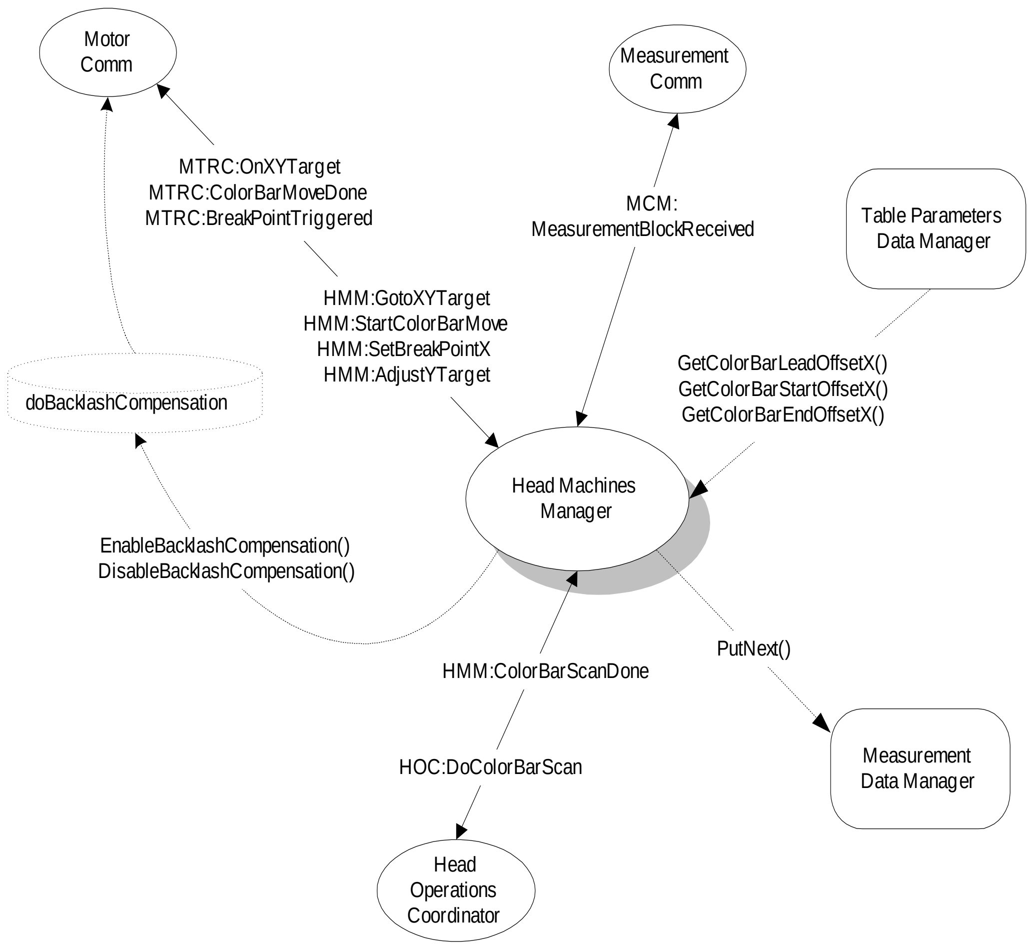
Note: For accuracy, a “Move Settle Wait” needs to be performed before the measurement. The first point measurement assumes that the Measurement Flash is ready.

HMM informs HOC each time a point is measured when performing this process.



## 6.14.2.1. HMM – System Interaction, DoColorBarScan

The diagram shows all the machines involved in performing the process.




As stated earlier, the Scan Color Bar is a special process. The state transition shown on the next page assumes that IMCSC takes care of directly sending the Measurement Command to the head module. After the reception of the measurement data, HMM simply goes back to the “Waiting for Head Reply” state if there is more point measurement data to be expected.

Also see Motor Chip Set Comm (IMCSC) Task Description on page 72 for more information about the Color Bar Scan process.

### 6.14.2.1.1. HMM – Do Color Bar Scan, State Transitions Level 1

The "DoColorBarScan" message is generated by HOC. This message is accompanied by data.

Data1 = scan Start Point Index
Data2 = scan End Point Index

Based on these 2 data, HMM queries SPDM to get the point coordinate parameters. The Start Offset is used to make sure that the probe head is already moving at a constant speed before the first color bar patch measurement is taken. The End Offset on the other hand is used to make sure that the last color bar patch measurement has already been taken before the deceleration begins.

The Lead Offset is a special parameter used to compensate for all the Color Smart System latencies that will be involved in the color bar scan process. The idea is simply to send the probe head "Measurement Command" a few steps before the actual point to be measured. This way, the coordinate at which the measurement will be made is more accurate (center of the color bar patch).

Note: HMM informs HOC each time a point is measured when performing this process.

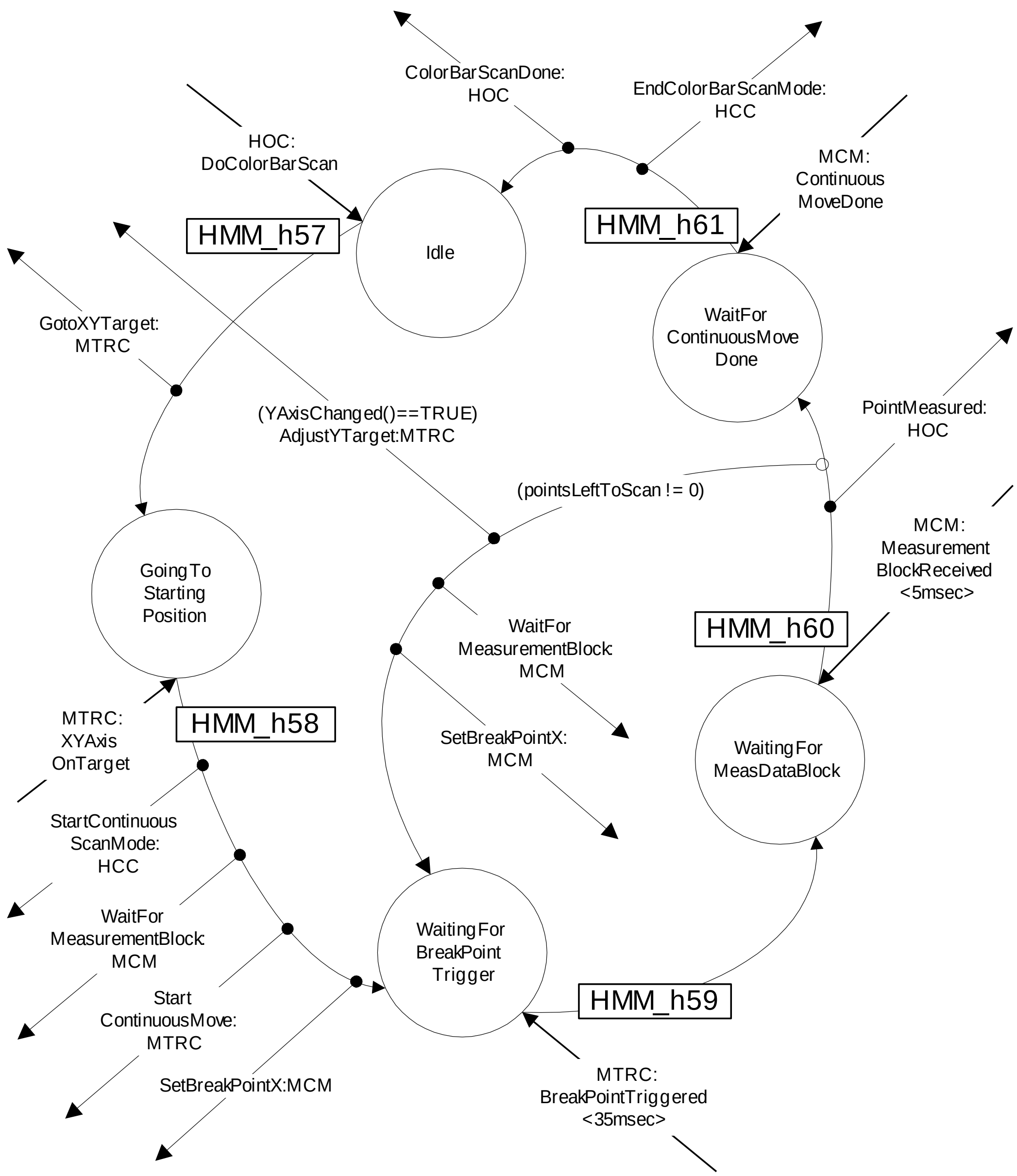
**IMPORTANT: This process assumes that the move is slow enough for the Measurement Flash to be fully recharged before the next breakpoint occurs. This is why there is no state for "Measurement Flash Recharge Wait" which is seen in the previous behavioral diagrams.**

**The motor chipset's break points are updated each time the measurement data block is received. The x-axis may also be adjusted if necessary.**

**The Backlash is compensated when the probe head moves to the Starting Position. The Backlash is then disabled throughout the continuous scan move.**

**6.14.2.1.2.** HMM – Do Color Bar Scan, State Transitions Level , continued . . .

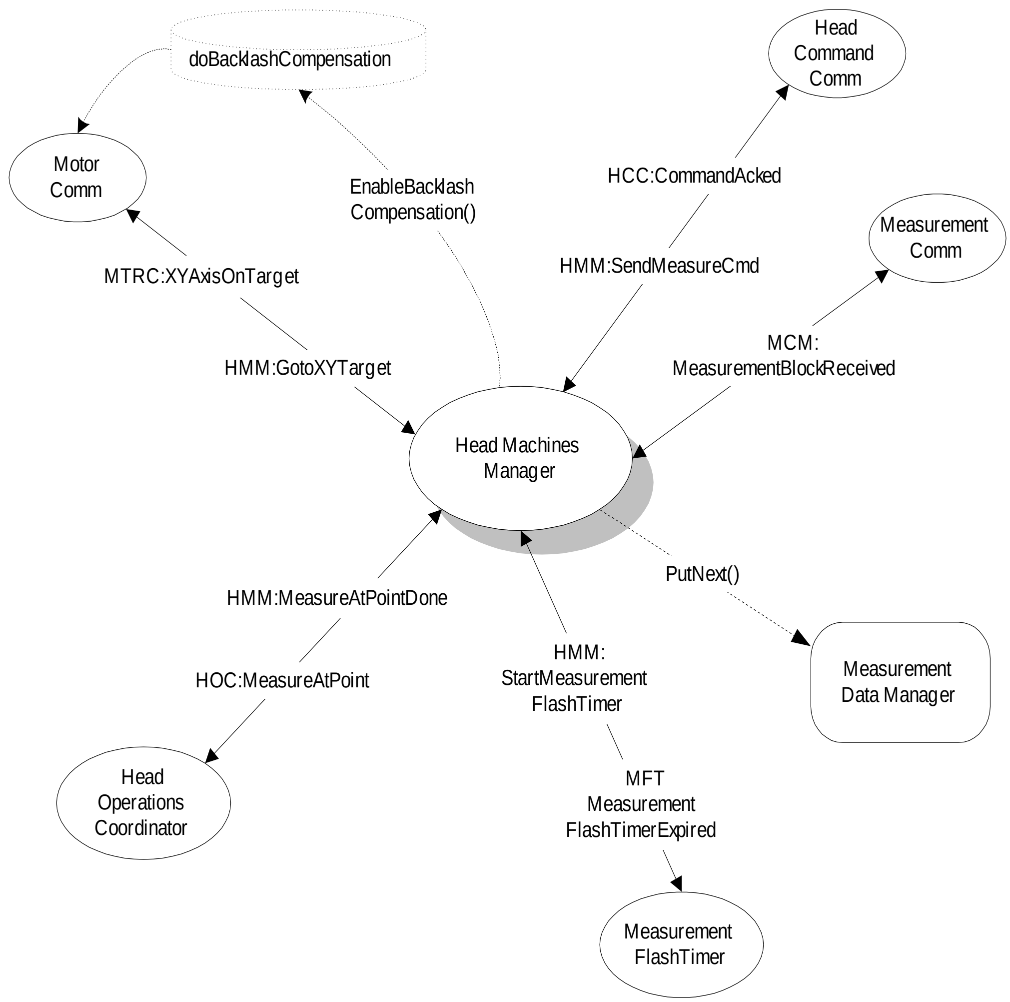
In general the state transition below shows that the probe head is initially positioned at the beginning of the color bar. The probe head is then sent towards the end of the color bar (left or right table direction). “Probe Head Command Acked” messages are received as the head moves along. This signals the soon-arriving measurement data. The data is collected and stored using the services of MDM.



### 6.14.2.2. HMM – System Interaction, MeasureAtPoint

NOTE: The probe head will be left at the location of the point being measured after this process is completed.

The diagram shows all the machines involved in performing the process.



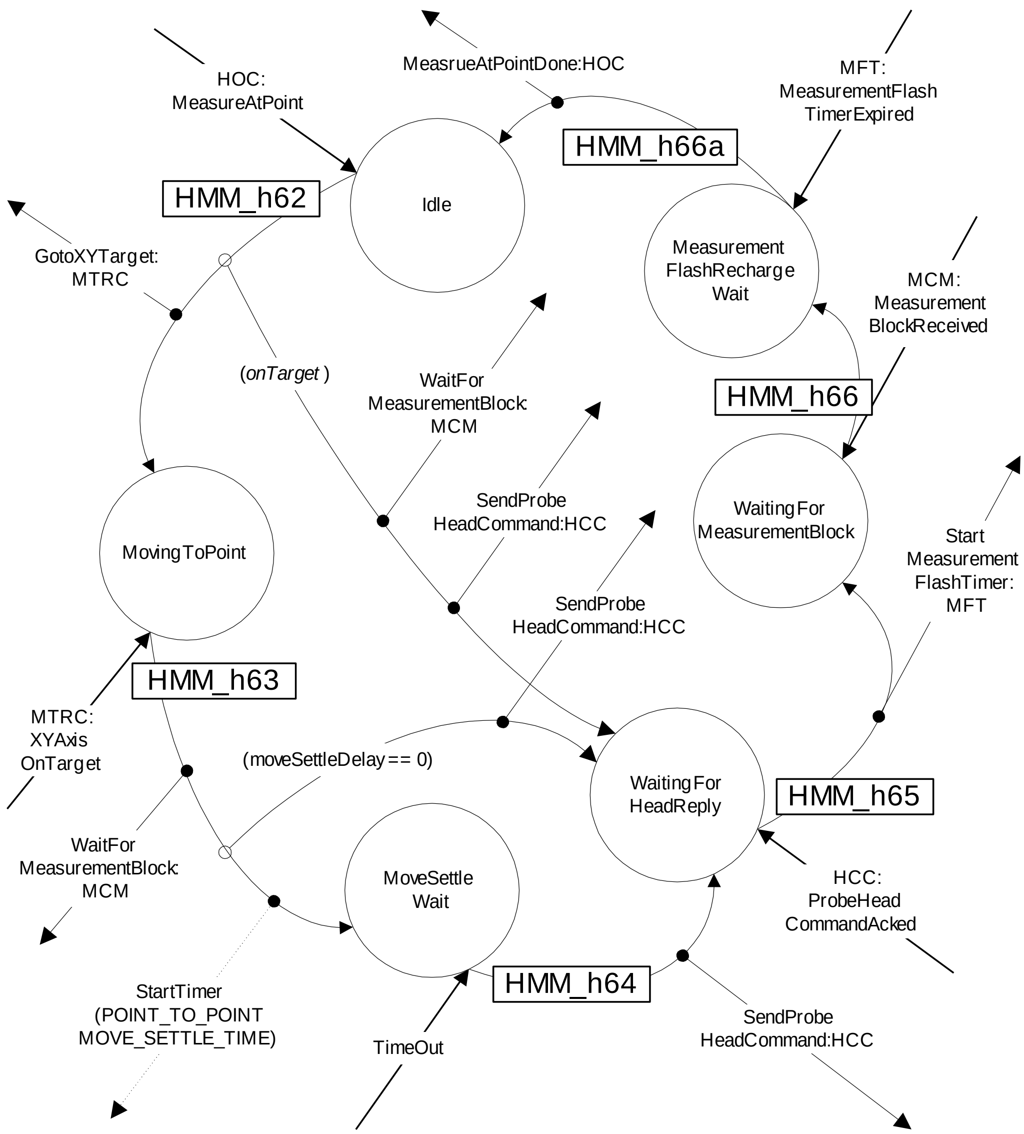

**IMPORTANT: CSM is responsible for "Resetting" MDM, before making this request. This procedure simply inserts the next measurement into the next available slot of MDM.**

### 6.14.2.2.1. HMM – Measure At Point, State Transitions

HOC generates the "MeasureAtPoint" message. As stated earlier, HOC is responsible making sure that the probe head is correctly set (Raised or Lowered), before sending this request.

This message is accompanied by data.

Data1 = X Point Coordinate
Data2 = Y Point Coordinate



### 6.14.2.3. Measurement Flash Timer (MFT) Task Description

***MFT – M**easurement **F**lash **T**imer*

This machine keeps track of the elapsed time since the Measurement Flash was last used. This allows HMM to ensure that the Measurement Flash has been given enough time to recharge before the next “Measurement” request is made to the probe head.

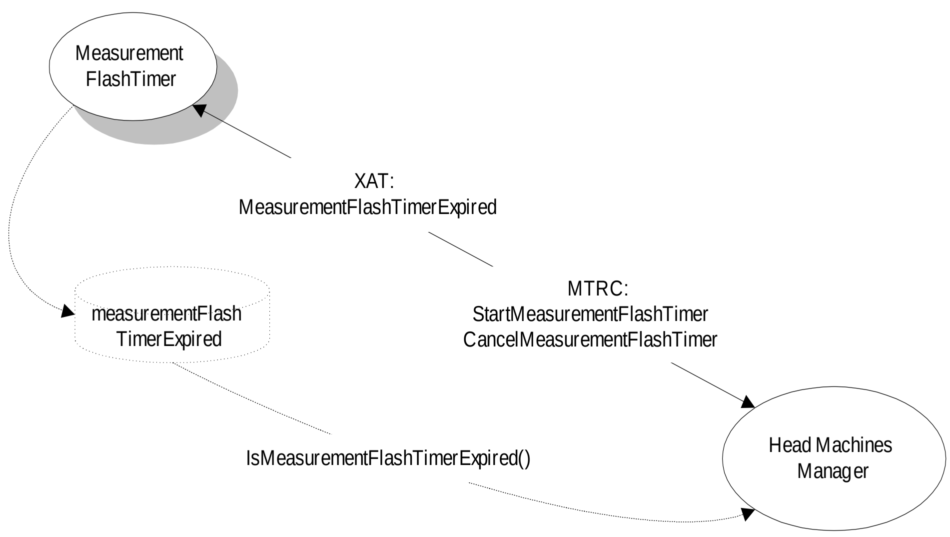
The “StartMeasurementFlashTimer” is a message request sent by HMM. This message is accompanied with data:

Data1 = Timeout Duration in milliseconds
Data2 = NULL

**IMPORTANT:**

**The timeout requested is subject to the Kernel’s timer limitations.**

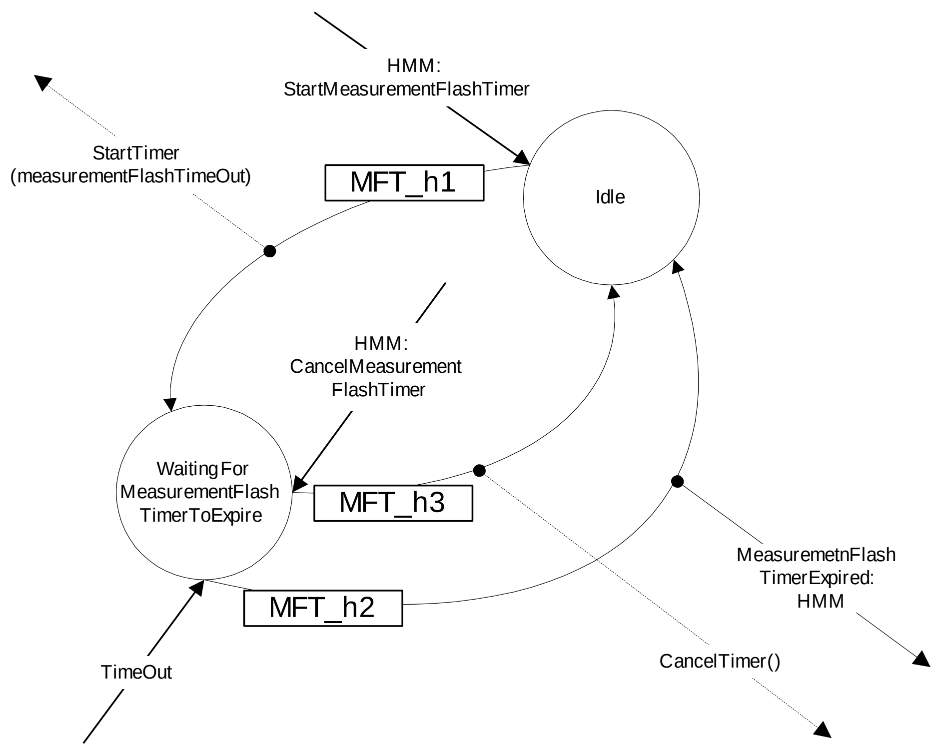
### 6.14.2.4. MFT – System Interaction, StartMeasurementFlashTimer, CancelMeasurementFlashTimer



### 6.14.2.4.1. MFT – StartMeasurementFlashTimer, State Transitions

MFT only has a few simple state transitions. When it receives a "StartMeasurementFlashTimer" message, it generates a Timer Request to the Kernel and then goes to a "Waiting" state. The Kernel sends a TimeOut message when the requested time has expired. MFT in turn tells HMM about the event.

Note: The timer request can also be cancelled.



### *6.15. Status LED (SLED) Task Description*

### ***S**LED - **S**tatus **LED***

This SM will handle the System LED for status displays. This SM will utilize timing services of the OS Kernel.

There are five different status LEDs provided by the Color Smart controller:

a) System Status LED
b) PCI Communication Status LED
c) Head Communication Status LED
d) Motor Chip Set Status LED
e) Measurement Data Status LED

## 6.1. Color Smart System Monitor Task Description

## 6.2. Scan Parameters Data Manager (SPDM) Task Description

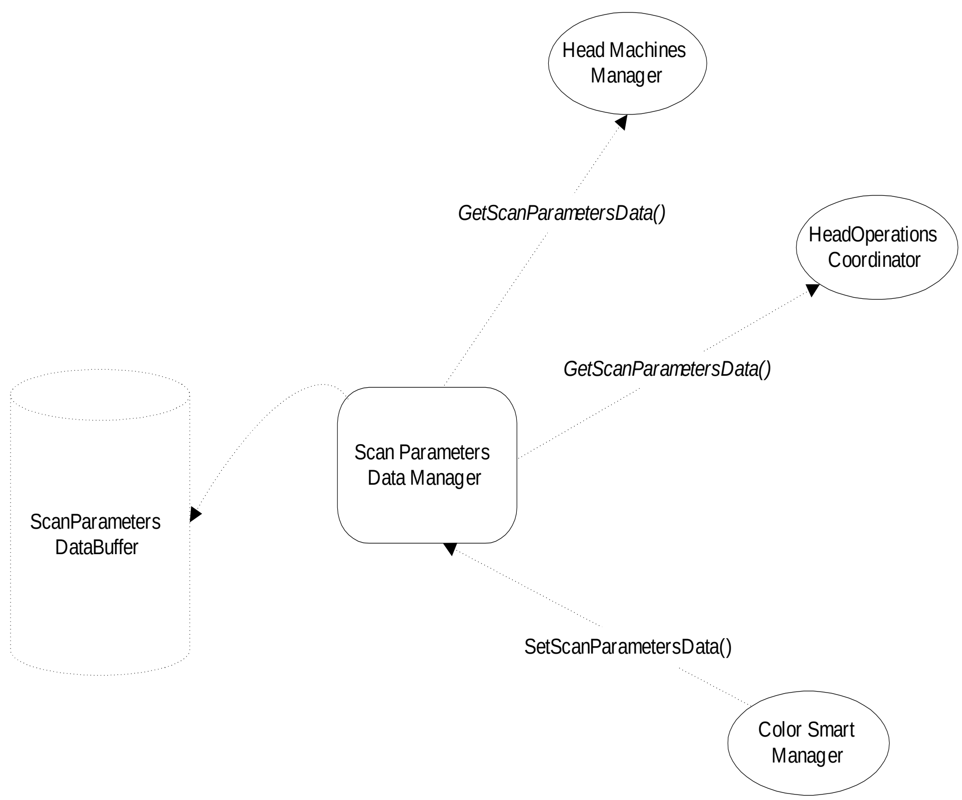
### SPDM - Scan Parameters Data Manager

This manager handles all the operation required for managing the Scan Coordinate as well as the Scan Type (Color Bar Flag) DataBase. Data access is accomplished through a set of interfaces provided by the manager. As the context diagram depicts, only the Color Smart System Manager should "request" the storage of a given Scan Coordinate or Scan Type data. The Color Smart NT App will provide these Data. The context diagram also shows that several SM - can query the database. The following SM can query the database: HMM, HOC, and CSM.

A scan coordinate data consists of the X and the Y target location as well as additional data for move generation (Move Profile). The X and Y target location is provided by the Color Smart NT. These target data contain absolute stepper motor "steps" which is used to send the motors to a specific location.

**IMPORTANT:**

**CSM checks to make sure that the size of the scan parameters is valid. SPDM can have a maximum of 1000 scan coordinate data. The validity of the data itself (coordinates etc) is checked only at the time when it is being used - when scanning for example.**



## *6.3. Table Parameters Data Manager (TPDM) Task Description*

***TPDM – T**able **P**arameters **D**ata **M**anager*

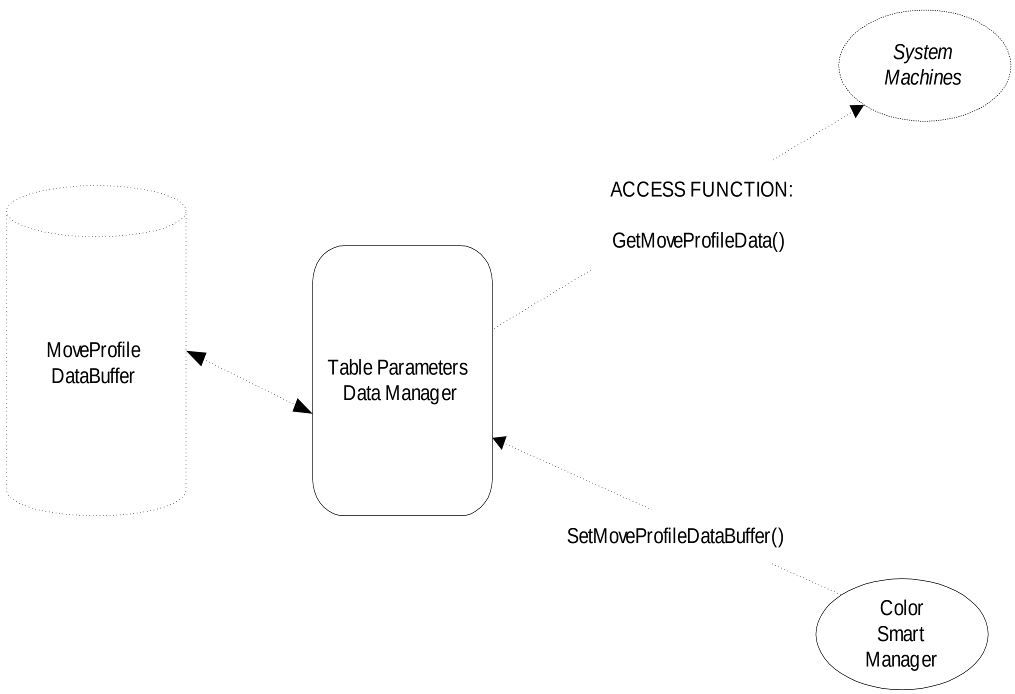
This manager holds all of the table parameters data needed for the CS operation.

**IMPORTANT:**

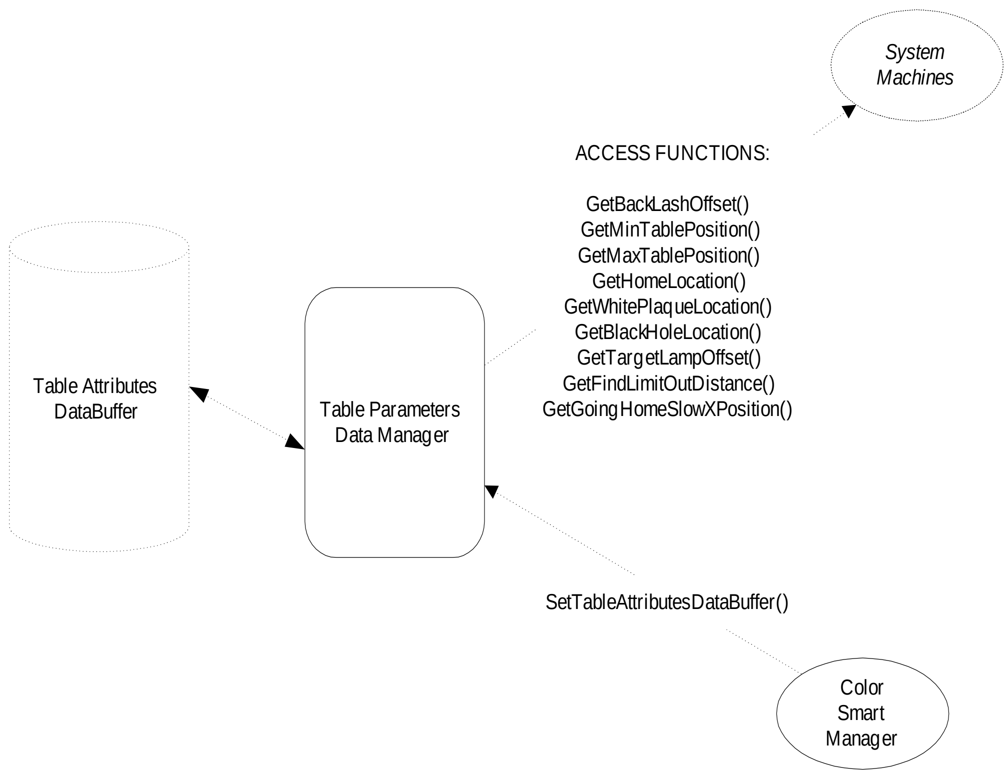
**All of the System Machines, who need data from this TPDM, create its own link to this manager. These System Machines make their data request data through the use of the manager's access services.**

**The NT Application is allowed to change all of these data (in certain modes/states). This implies that the data can either be "end user configurable" or "factory configurable".**

### 6.3.1. Move Profile Data Access



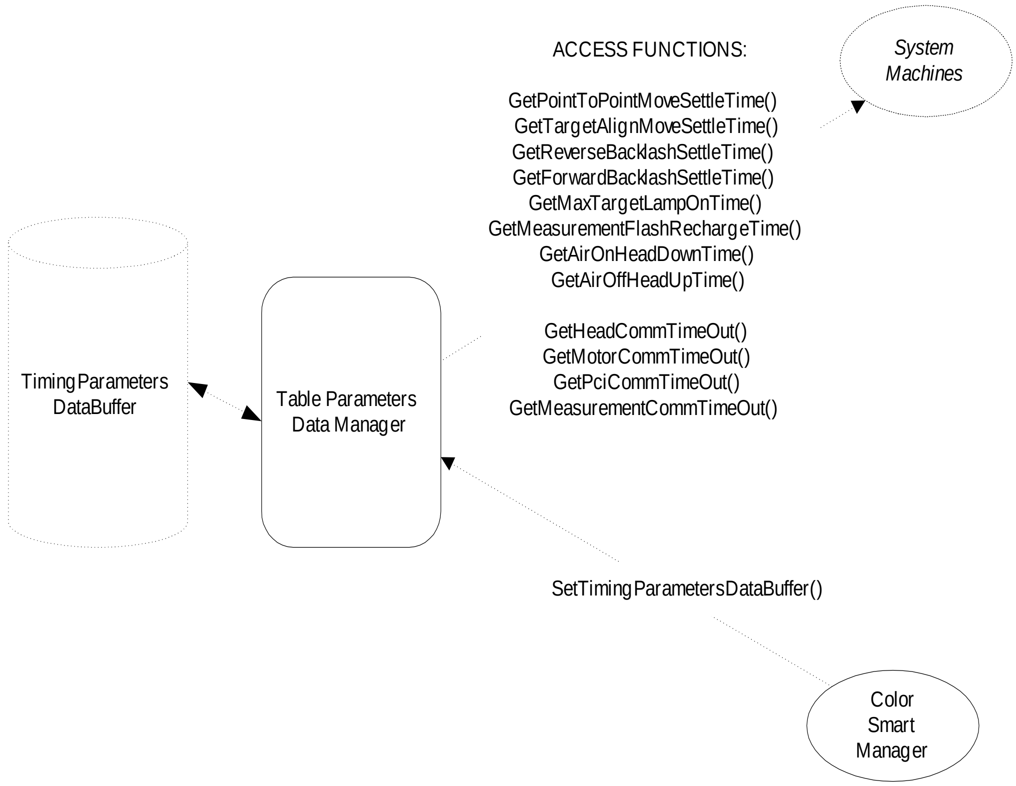
## 6.3.2. Table Attributes Data Access



Below is a short description of the different Table Attributes variables

| **Variable Name** | Description |
|---|---|
| **Backlash Offset** | Backlash parameter (absolute value). The intermediate target position before moving to the final position. |
| **Min Table Position** | The minimum table coordinate. This is used to validate that the coordinates of the requested moves (scanning or trackball moves) is within the valid range. |
| **Max Table Position** | The maximum table coordinate. This is used to validate the coordinates of the requested moves is within the valid range. |
| **Home Location** | This is the home coordinate location on the table |
| **White Plaque Location** | This is the White Plaque coordinate location on the table |
| **Black Hole Location** | This is the Black Hole coordinate location on the table |
| **Target Lamp Offset** | This is the Target Lamp Center offset with respect to the Center of the measurement sensor. (absolute value) |
| **Find Limit Out Distance** | This is the distance used when initially moving away from the limits during “Find Limits” procedure. (absolute value) |
| **Going Home Slow X Position** | This is the X axis coordinate on the table where the probe head begins to move slow during the “Go Home” procedure. |

### 6.3.3. Timing Parameters Data Access




Below is a short description of the different Timing Parameter variables

| **Variable Name** | Description |
|---|---|
| **Point To Point Move Settle Time** | This is the delay made after the final Point To Point move is done before measurement command is sent to the Probe Head. This operation is used to settle the Ringing of the Step Motors. |
| **Target Align Move Settle Time** | This is the delay made after a Target Align Step is done before measurement command is sent to the Probe Head. This operation is used to settle the Ringing of the Step Motors. |

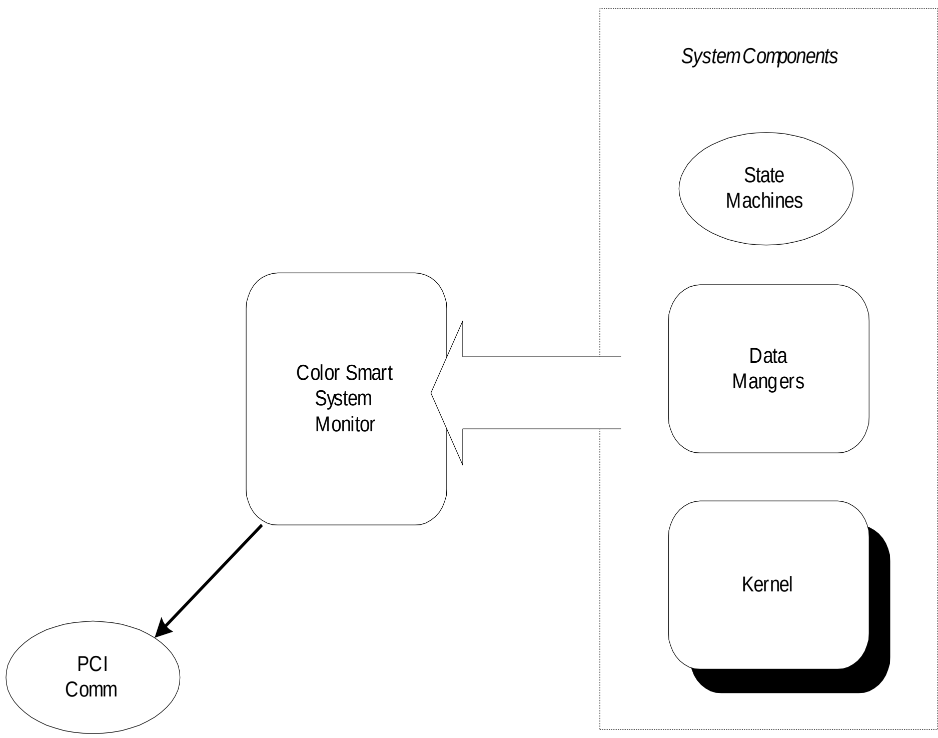
| Variable Name | Description |
| --- | --- |
| **Reverse Backlash Settle Time** | This is a delay used to ensure that the Step motors will be ready in case it has to make a reverse move. This is used to avoid Stepper Motor slippage.<br><br>**IMPORTANT: This delay is assumed to be greater than the forward delay. There are sections in the code that uses this assumption.** |
| **Forward Backlash Settle Time** | This is a delay used to ensure that the Step motors will be ready to make another forward move (same axis direction). This is used to avoid Stepper Motor slippage. |
| **Max Target Lamp On Time** | This is the maximum time that the Target Lamp is allowed to stay On. The Controller card automatically turns it Off after this time elapses. |
| **Measurement Flash Recharge Time** | This is the time it takes for the Measurement Flash to recharge. The controller card uses this to make sure it waits before sending the next Measure Command to the Probe Head.<br><br>**IMPORTANT: This delay is assumed to be greater than the Reverse Backlash Settle Time. There are sections in the code that uses this assumption.** |
| **Air On Head Down Time** | This is the time it takes for the Probe Head to be fully lowered on the table surface (bouncing settled) |
| **Air Off Head Up Time** | This is the time it takes for the Probe Head to be fully raised from the table surface |
| **Head Comm Time Out** | This is the maximum time the Controller waits for the reply to a command before considering it a failure. |
| **Motor Comm Time Out** | This is the maximum time the Controller waits for the motor chipset to be ready from the last communication before considering it a failure. |
| **PCI Comm Time Out** | This is the maximum time the Controller waits for the Mail Box out to be available before considering it a failure. |
| **Measurement Comm Time Out** | This is the maximum time the Controller waits for the measurement data (512 Bytes) before considering it a failure. |

## *6.4. Color Smart System Monitor (CSSM) Task Description*

## ***CSSM – C****olor* ***S****mart* ***S****ystem* ***M****onitor*

This manager simply provides all the connection to all of the State Machines within the system. It will provide all the interfaces that will be necessary for diagnostics or debugging purposes.

Through the CSSM, the NT Application has the ability to query any thing it wants to know about any part of the system.



# 7. PCI Communication Protocol

This section describes the protocol used between the Color Smart PCI Controller board and the Color Smart NT Application.

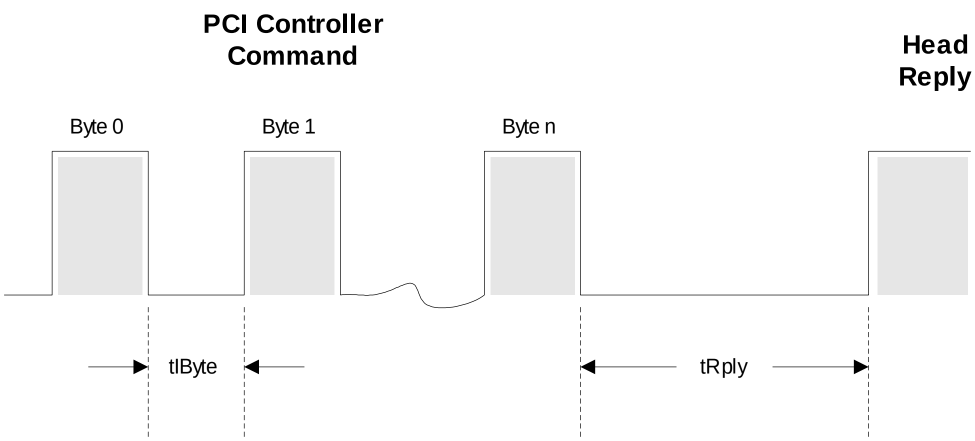
Command ID 2000 and up are used for system queries. There is no ACK involved with these commands. The reply will consist of the data requested.

A Command ID below 2000 is used for system operations including parameter settings.

1. System operations will reply with an ACK followed by a COMPLETION reply at a later time.
2. Parameters settings will reply with an ACK without the COMPLETION reply

# 8. Head Communication Protocol

## 8.1. Head Commands




RS232 , 8 Data, 1 Start Bit, 1 Stop Bit, 62.5 KBaud

| Label | approx. Time in milliseconds |
|---|---|
| tIByte | 1 bit time |
| tRply | 2 msec |
| | |
| 1 Bit Time | 16 usec |
| 1 Byte Time + 1 start + 1 stop | 160 usec |
| 4 Byte Transfer | 640 usec |

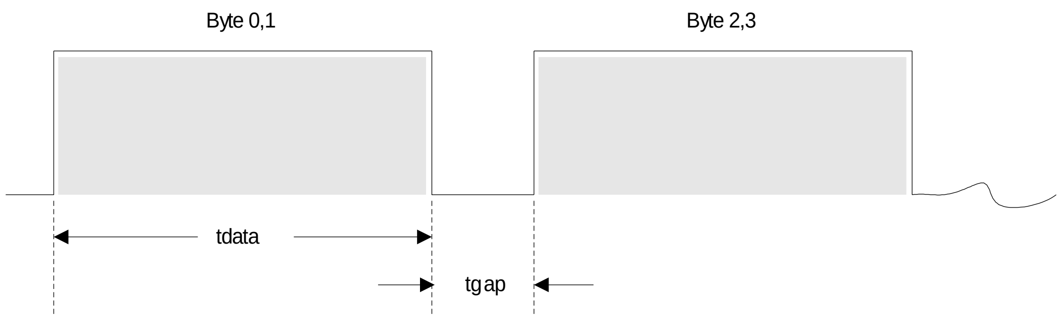
**IMPORTANT:**

**The CS Controller uses the reply to the “Measurement Command” as a timing reference to determine when the next flash or measurement can be initiated.**

## 8.2. Head Measurement

Each point measurement is 512 Bytes Long.

**Serial Data Timing**




| Label | Time |
|---|---|
| tdata | 16.0 usec |
| tgap | 4.0 usec |
| | |
| Total Transfer Time per WORD | 20.0 usec |
| | |
| Total Transfer Time per MEASUREMENT | 5.12 msec |
| | |
| Data Clock Freq | 1.0 Mhz |
| | |
| Throughput | |

# 9. Hardware

This section contains an overview of the Color Smart Hardware.

## *9.1. Interrupt Definitions*

The 386Ex processor contains 2 interrupt controller peripherals. One is configured as the master and the other a slave. These controllers are very similar to the ones found on your ordinary PC. The on-chip controllers support a total of 16 Interrupt sources. Cascading an external interrupt controller can expand the number of interrupt inputs.

The table below shows how the Master Interrupt Controller uses each of the eight interrupt inputs:

| Master Interrupt Controller | Interrupt Vector | Usage Description |
|---|---|---|
| IR0 | IRQ0 | Timer 0 produces the system tick used for timing services. |
| IR1 | IRQ1 | This is connected to the external interrupt input called INT0. INT0 is an input for the Motor Controller Chipset. The Motor Controller Chipset generates interrupts for changes of its status. |
| IR2 | IRQ2 | The 2 interrupt controllers are cascaded internally within the 386Ex. IR2 is used as an input for the Slave Interrupt Controller |
| IR3 | IRQ3 | **Unused.** IR3 is internally dedicated as an input for COM2 (Asynchronous Serial Port 1). The PCI Controller only needs one communication port. |
| IR4 | IRQ4 | The PCI Controller talks to the head through serial communication. IR4 is internally used by COM1 (Asynchronous Serial Port 0) as the interrupt input. |
| IR5 | IRQ5 | This is connected to the external interrupt input called INT1. INT1 is an input for the PCI Bridge. The PCI Bridge generates interrupts when the “mailbox” register has new data. |
| IR6 | IRQ6 | **Unused.** The external interrupt input is configured as a general-purpose output. |
| IR7 | IRQ7 | **Unused**. The external interrupt input is configured as a general-purpose output. |

The table below shows how the Master Interrupt Controller uses each of the eight interrupt inputs:

| Slave Interrupt Controller | Interrupt Vector | Usage Description |
|---|---|---|
| IR0 | IRQ8 | **Unused.**<br>This can be connected to the external interrupt input called INT4. This port is not used. |
| IR1 | IRQ9 | **Unused.**<br>IR1 is internally dedicated as an input for the Synchronous Serial Port (SSIO). Due to high-speed data transfers on the SSIO port, the DMA feature is utilized instead. This means that the PCI Controller does not execute an interrupt service for every data that the SSIO receives. An interrupt is generated by the DMA only when the full “data block” is received. |
| IR2 | IRQ10 | **Unused.**<br>IR2 is internally dedicated as an input for Timer 1. This peripheral is not used. |
| IR3 | IRQ11 | **Unused.**<br>IR3 is internally dedicated as an input for Timer 2. This peripheral is not used. |
| IR4 | IRQ12 | This input is directly connected to the DMA Controllers (DMA0 and DMA1). See description of IR1 for more details. |
| IR5 | IRQ13 | **Unused.**<br>This can be connected to the external interrupt input called INT6. This port is not used. |
| IR6 | IRQ14 | **Unused.**<br>This can be connected to the external interrupt input called INT7. This port is not used. |
| IR7 | IRQ15 | This is an interrupt input for WatchDog peripheral. |

## *9.2. IO Port Definitions*

This section shows how each of the General Purpose IO ports are being utilized. The highlighted boxes indicate unused and available IO ports.

### 9.2.1. PORT 1 Definition

| Port Number | Usage Description |
|---|---|
| P1.0 | **General Purpose Output.**<br>Used for the Table Air Control. |
| P1.1 | **General Purpose Output.**<br>Used as the Head Reset Control. |
| P1.2 | **General Purpose Input.**<br>Used to detect Motor Chipset HOST READY line. |
| P1.3 | **General Purpose Output.**<br>Used for Motor Power ON/OFF control.<br>P1.3 is multiplexed with DSR for Serial Port 0, which is disabled. |
| P1.4 | **General Purpose Input.**<br>Used as the PCI Mailbox Out status flag<br>P1.4 is multiplexed with RI (Ring Indicator) for Serial Port 0, which is disabled. |
| P1.5 | **Unused**<br>**LOCK (peripheral mode)**<br>P1.5 is multiplexed and disabled. |
| P1.6 | **Used as HOLD (peripheral mode)**<br>For the shared memory design this signal is used to perform bus arbitration with the PCI Bridge.<br>P1.6 is multiplexed and disabled. |
| P1.7 | **Used as HOLDA (peripheral mode)**<br>For the shared memory design this signal is used to perform bus arbitration with the PCI Bridge.<br>P1.7 is multiplexed and disabled. |

## 9.2.2. PORT 2 Definition

| Port Number | Usage Description |
|---|---|
| P2.0 | **Used as Chip Select 0 (CS0).**<br>CS0 in conjunction with the BLE and BHE lines, selects bank 0 and bank 1 of the Local System Ram.<br>P2.0 is multiplexed and disabled. |
| P2.1 | **Used as Chip Select 1 (CS1).**<br>CS1 selects the Motor Chipset Controller when data is being passed back and forth to it.<br>P2.1 is multiplexed and disabled. |
| P2.2 | **Used as Chip Select 2 (CS2).**<br>CS2 selects the Auxiliary input ports.<br>P2.2 is multiplexed and disabled. |
| P2.3 | **Used as Chip Select 3 (CS3).**<br>CS3 selects the Auxiliary output ports.<br>P2.3 is multiplexed and disabled. |
| P2.4 | **Used as Chip Select 4 (CS4).**<br>CS4 selects the PCI Bridge’s MailBox IN.<br>P2.4 is multiplexed and disabled. |
| P2.5 | **Used Asynchronous Serial Communication RX.**<br>**Serial Port 0.**<br>This port is used to communicate (receive replies) from the head module.<br>P2.5 is multiplexed and disabled. |
| P2.6 | **Used Asynchronous Serial Communication TX.**<br>**Serial Port 0.**<br>This port is used to communicate (send commands) to the head module.<br>P2.6 is multiplexed and disabled. |
| P2.7 | **Used as Motor Chipset Reset.**<br>This port is used to perform a hard reset of the motor chipset.<br>P2.7 is multiplexed with CTS (Clear To Send) Serial Port 0, which is unused. |

### 9.2.3. PORT 3 Definition

| Port Number | Usage Description |
|---|---|
| P3.0 | **Unused.**<br>P3.0 multiplexed with Timer 0 OUT and is disabled. |
| P3.1 | **Unused.**<br>P3.1 multiplexed with Timer 1 OUT and is disabled. |
| P3.2 | **Used as INT0 interrupt input.**<br>The Motor Chipset Controller uses INT0 as an interrupt input to indicate the status changes within the chipset. P3.2 multiplexed and is disabled. |
| P3.3 | **Used as INT1 interrupt input.**<br>The PCI Bridge uses INT1 as an interrupt input to indicate the presence of new data in the mailbox. P3.3 multiplexed and is disabled. |
| P3.4 | **Unused.**<br>Pin is multiplexed with INT2. |
| P3.5 | **Unused.**<br>Pin is multiplexed with INT3. |
| P3.6 | **Unused.**<br>Pin is multiplexed with PWRDOWN. |
| P3.7 | **Used as COMCLK.**<br>An external CMOS oscillator is connected to provide a 16Mhz reference. This is so the controller card can generate a Baud Rate of 62.5 K, which is required to communicate with the Probe Head. |

### 9.2.4. Auxiliary OUTPUT PORT Definition

An auxiliary output port is provided to expand the existing outputs of the 386ex. Below is a description of how each of the auxiliary output ports is used:

| Port Number | Usage Description |
|---|---|
| AO.0 | System Status LED |
| AO.1 | PCI Status LED |
| AO.2 | Head Comm Status LED |
| AO.3 | Motor Chipset Status LED |
| AO.4 | Head Measurement Status LED |
| AO.5 | *Unused* |
| AO.6 | *Unused* |
| AO.7 | *Unused* |
| AO.8 | *Unused* |
| AO.9 | *Unused* |
| AO.10 | *Unused* |
| AO.11 | *Unused* |
| AO.12 | *Unused* |
| AO.13 | *Unused* |
| AO.14 | *Unused* |
| AO.15 | *Unused* |

### 9.2.5. Auxiliary INPUT PORT Definition

An auxiliary input port is provided to expand the existing inputs of the 386ex. Below is a description of how each of the auxiliary input ports is used:

| Port Number | Usage Description |
|---|---|
| AI.0 | +48 Volt, Motor Driver power supply status |
| AI.1 | +12 Volt, Interconnect board and head power supply status |
| AI.2 | +5 Volt, Interconnect board power supply status |
| AI.3 | *Unused* |
| AI.4 | *Unused* |
| AI.5 | *Unused* |
| AI.6 | *Unused* |
| AI.7 | *Unused* |
| AI.8 | *Unused* |
| AI.9 | *Unused* |
| AI.10 | *Unused* |
| AI.11 | *Unused* |
| AI.12 | *Unused* |
| AI.13 | *Unused* |
| AI.14 | *Unused* |
| AI.15 | *Unused* |

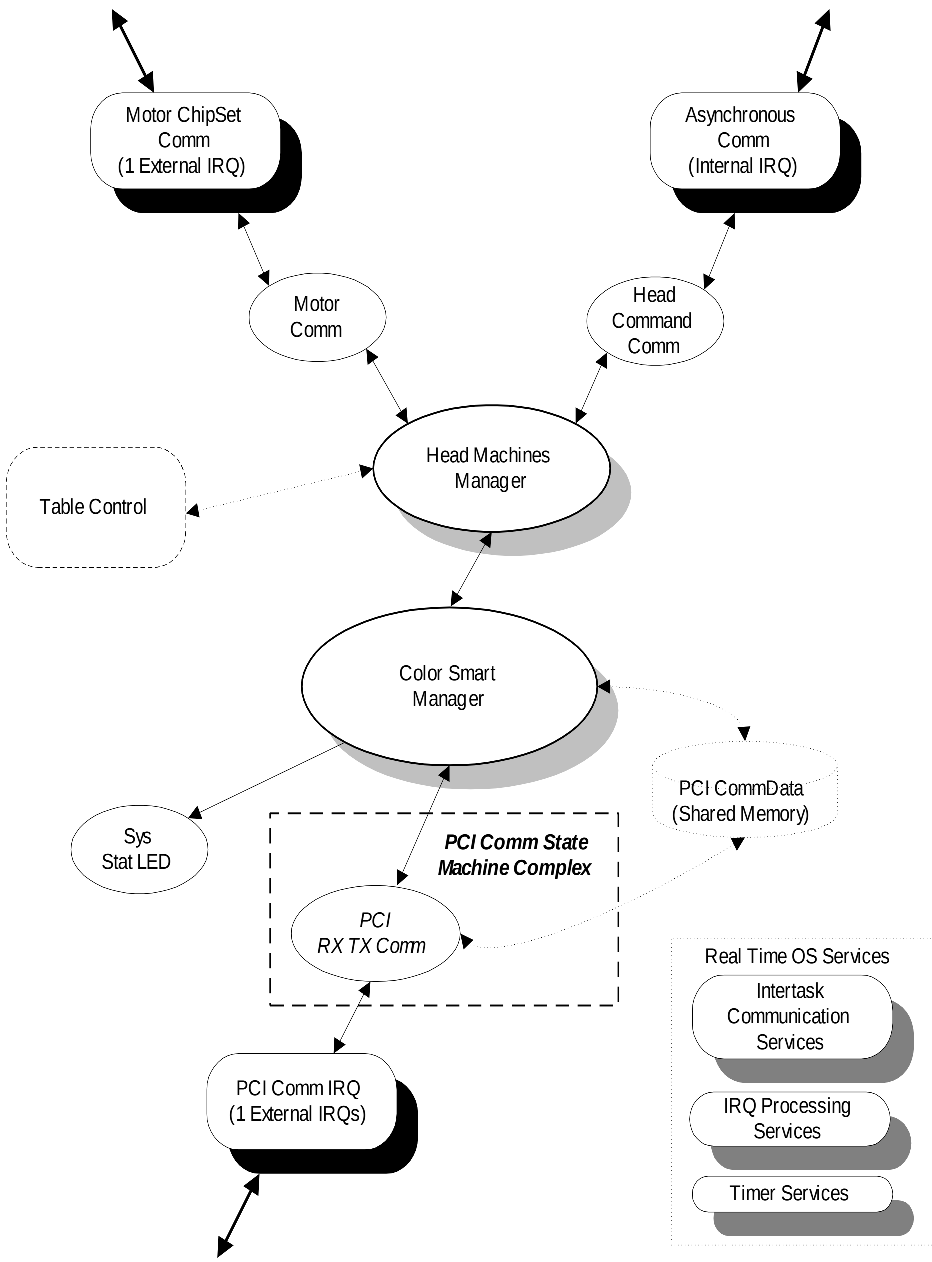
## *9.3. Chip Select Definitions*

This section shows how each of the Chip Select Lines are being utilized.

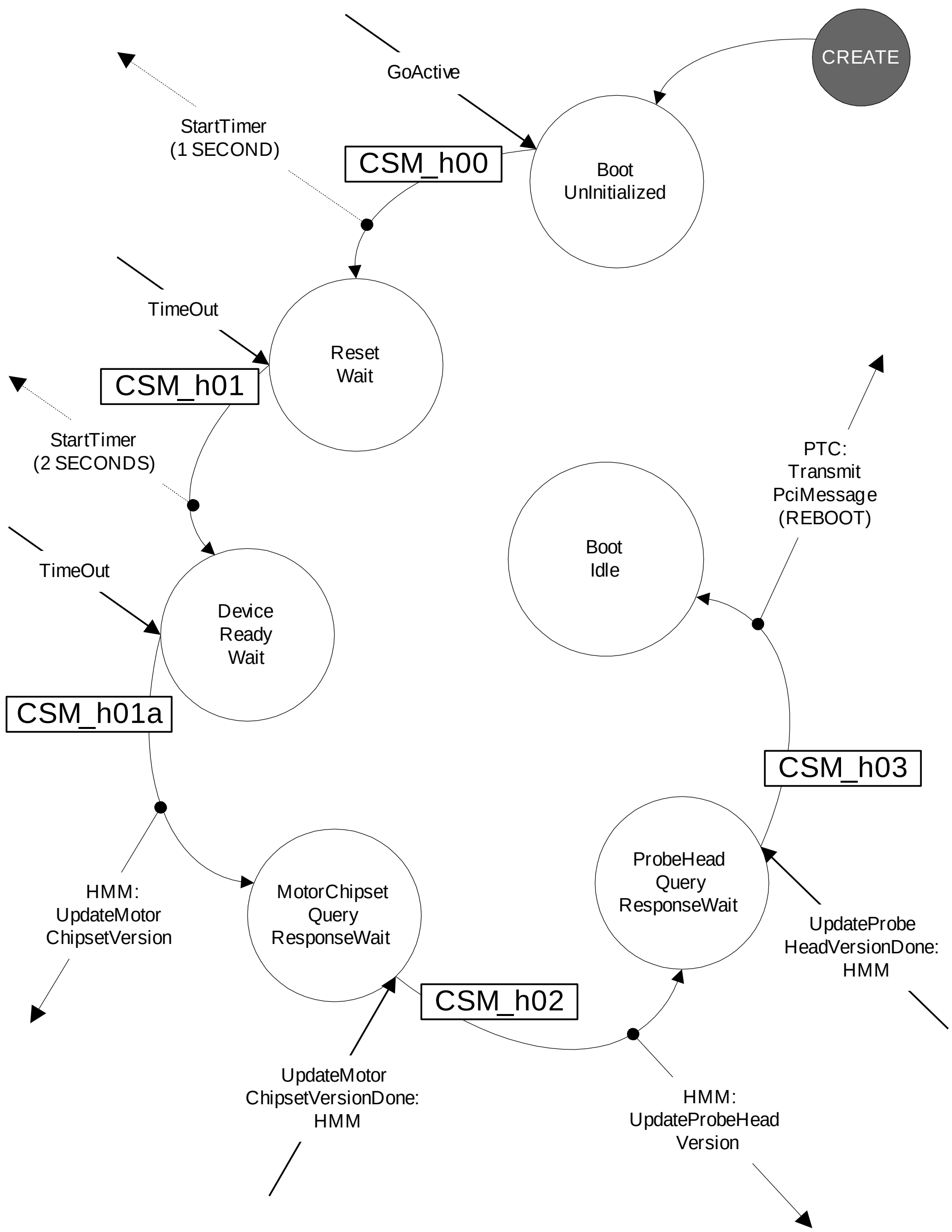
| Chip Select No | | | | | | |
|---|---|---|---|---|---|---|
| | Mem | IO | Wait States | Bus Size | Range Used | Description |
| UCS | **X** | | 15 | 8 | - 0x03FFFFFF | 64 K BOOT ROM |
| CS0 (+BLE) | **X** | | 1 | 16 | 0x00 - 0x0007FFFF | 512K x 16<br>LOCAL STATIC RAM<br>(BANK 0) |
| CS0 (+BHE) | **X** | | 1 | 16 | 0x00080000 - 0x000FFFFF | 512K x 16<br>LOCAL STATIC RAM<br>(BANK 1) |
| CS1 | | **X** | 2 | 8 | 0x300 - 0x301 | Motor Control<br>Chip Set |
| CS2 | **X** | | 1 | 16 | 0x300000 – 0x3003FF | Auxiliary Input<br>(Memory Mapped) |
| CS3 | **X** | | 1 | 16 | 0x400000 – 0x4003FF | Auxiliary Output<br>(Memory Mapped) |
| CS4 | **X** | | 1 | 16 | 0x500000 - 0x5003FFF | PCI Mail Box IN |
| CS5 | **X** | | 1 | 16 | 0x600000 - 0x6003FFF | PCI Mail Box OUT |
| CS6 | | | | | | *Unused* |

# 10. The Boot Firmware Context Diagram

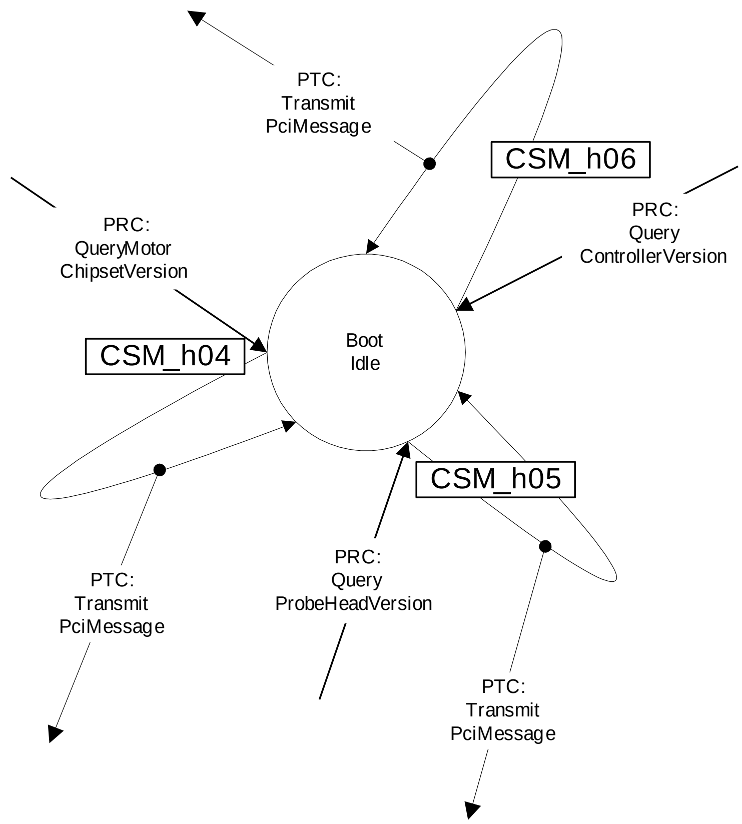
The Boot Firmware is a “stripped-down” version of the main Color Smart Application Firmware. Below is High Level Context diagram that shows the remaining components that are required by the Boot Kernel to operate.



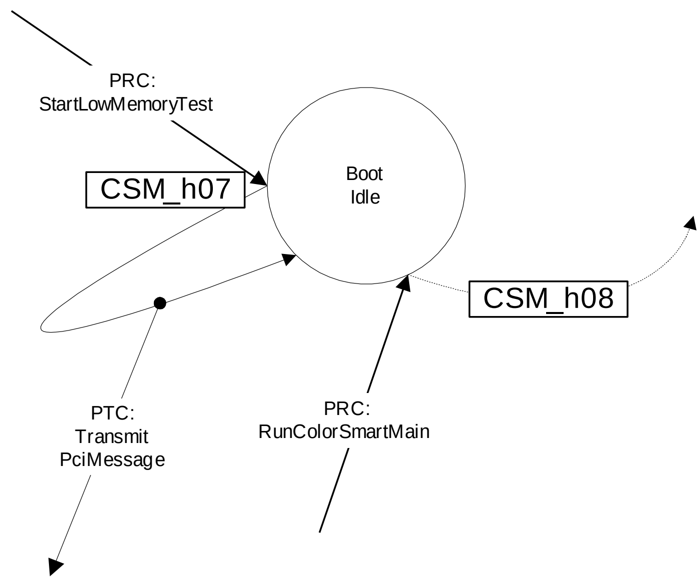
## 10.1. CSM Initial State Transitions



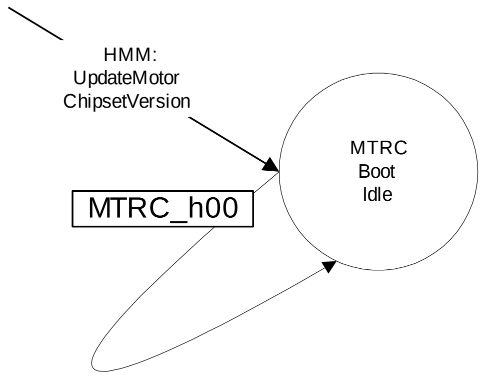
## 10.2. CSM Boot Idle State Transitions 1



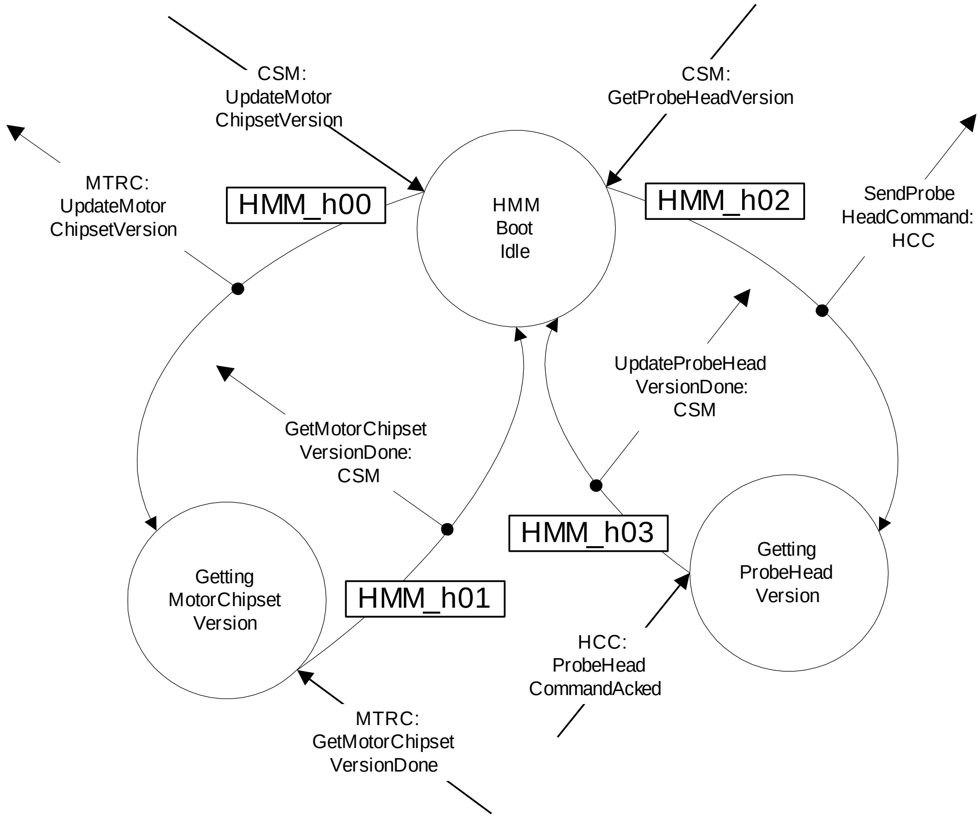
## 10.3. CSM Boot Idle State Transitions 2



## 10.4. MTRC Boot Idle State Transitions 1

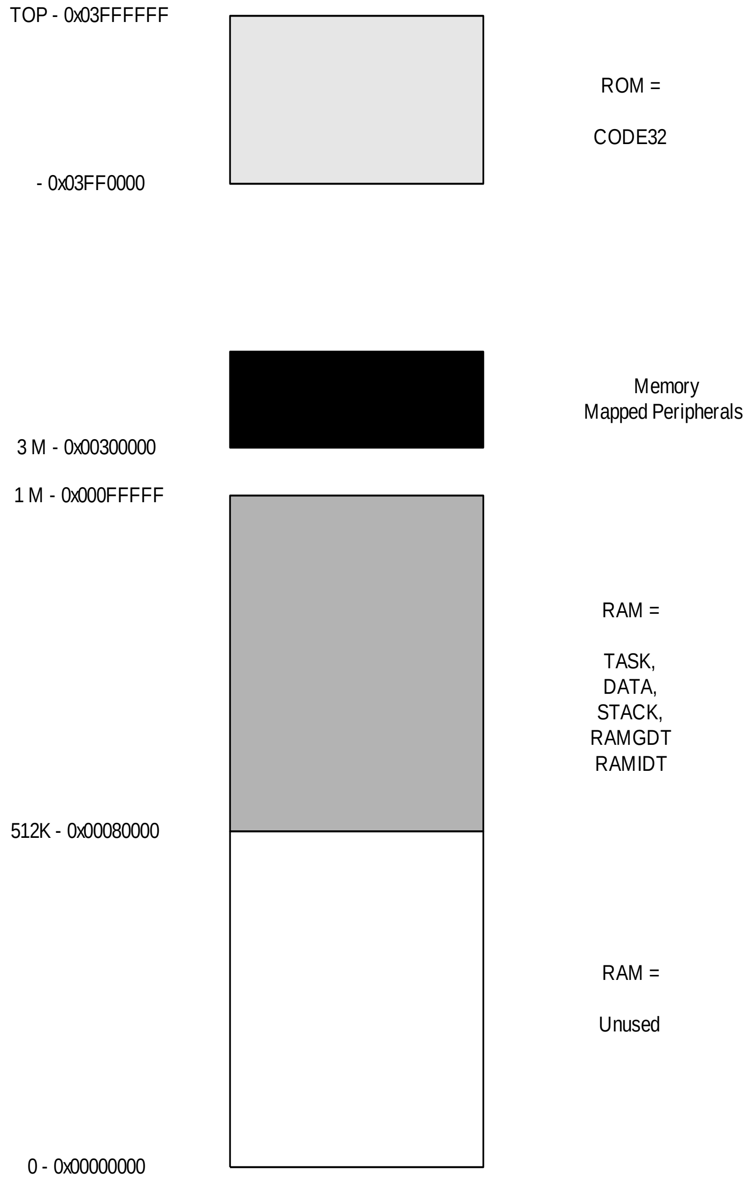


## *10.5. HMM Boot Idle State Transitions 1*

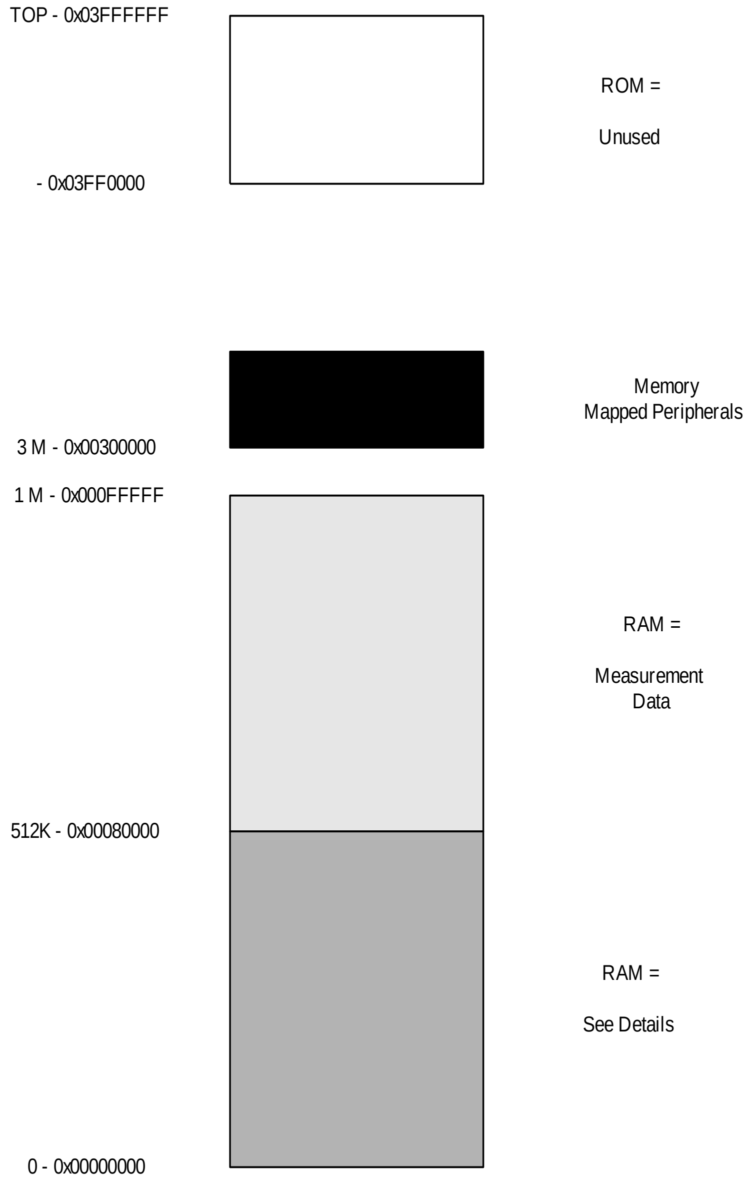


# 11. Memory Maps

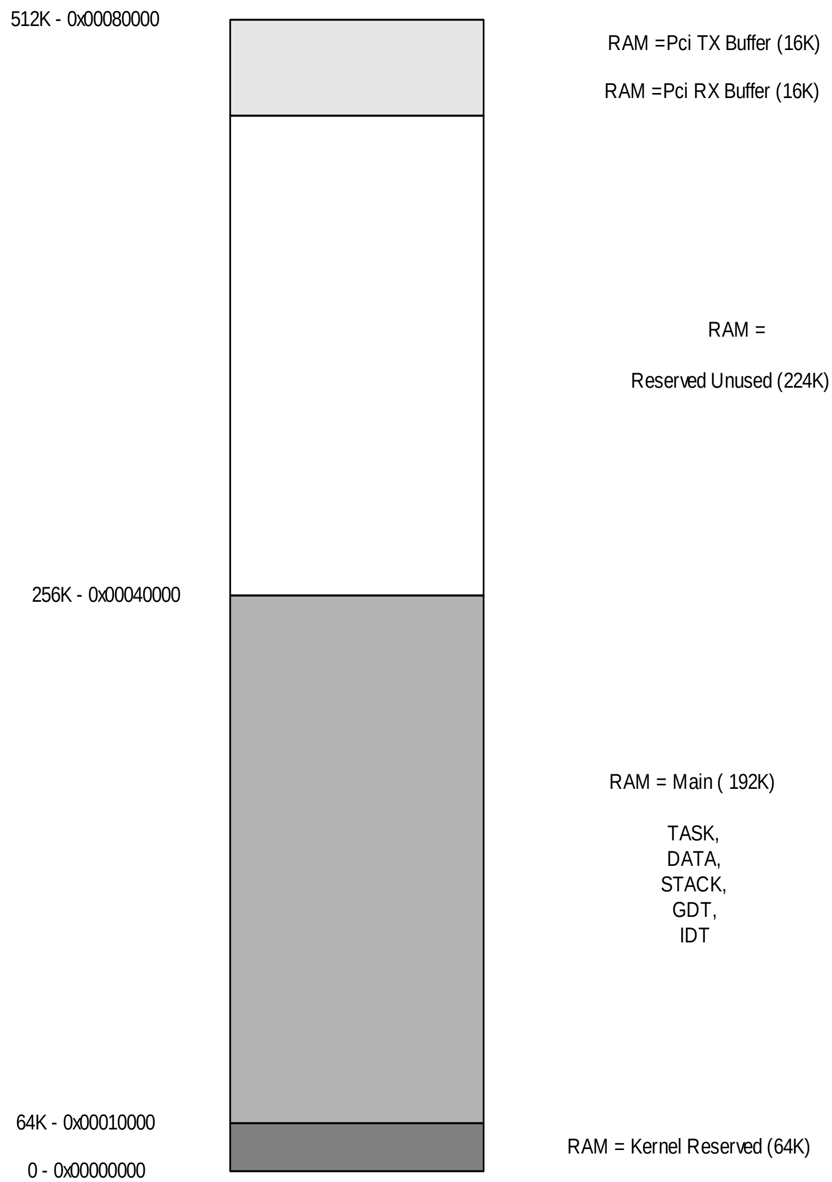
## *11.1. Boot Kernel Memory Map*



## 11.2. Color Smart Main Firmware Memory Map



## 11.3. Color Smart Main Firmware Memory Map, Details of the Lower 512 K Section



# 12. The Color Smart Table

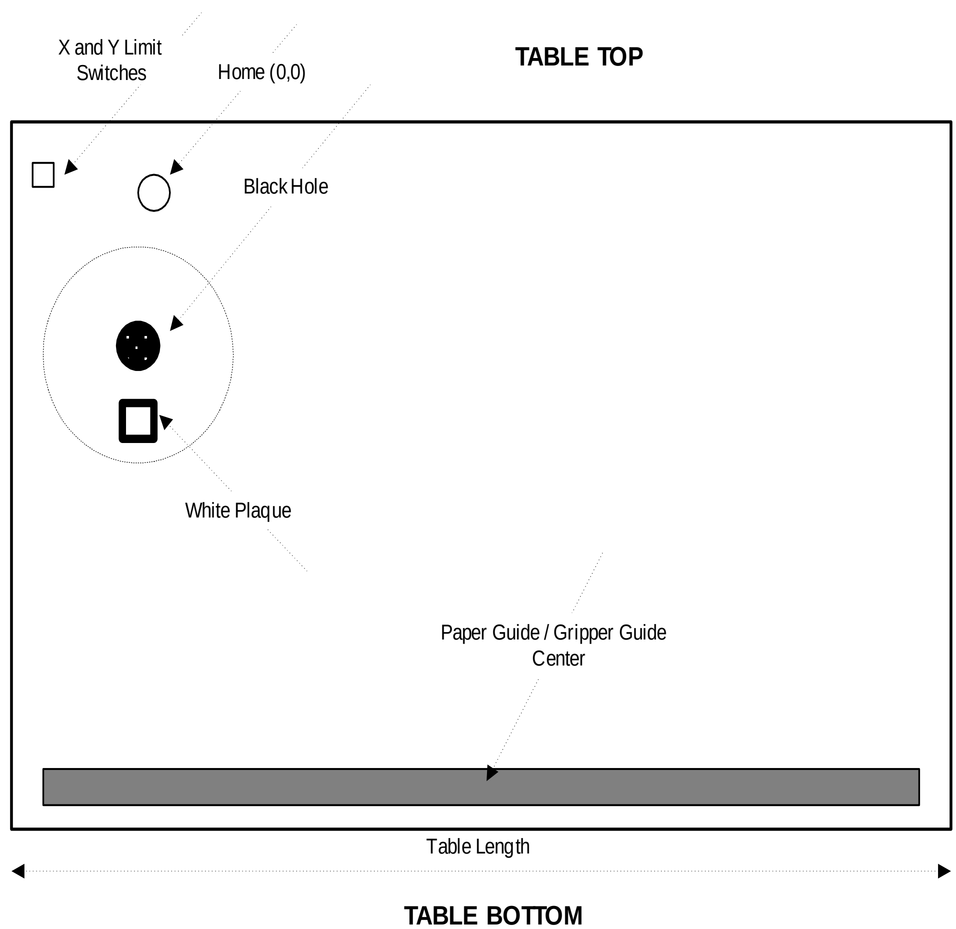
The Color Smart table is similar to the Auto Smart Table (AS10) with only a few differences. Below is a basic description of the table:

The table has a White Plaque on the upper left corner. This is used as a reference for calibration. The motor controller treats the center of the White Plaque as the X = 0 and Y = 0 position reference.

Specific to the Spectral Measurement version is the Black Hole located just below the White Plaque. This is used as a calibration reference as well.

The X and Y Home limit switches located at about 1 cm in the Y direction and about 3 cm in the X direction (away from the White Plaque). These home limit switches provides the mechanical starting point for the motor controller.

The table comes in a few different lengths depending on the customer's needs.

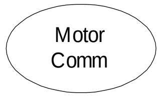


# 13. Motor Steps Conversion Table

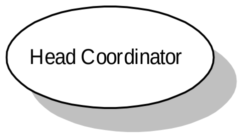
The table below gives us an idea of the distance the probe head travels with the different given motor steps.

| Motor Steps | centimeter | millimeter | inches |
|---|---|---|---|
| 1 | 0.00639 | 0.0639 | |
| 4 | 0.02556 | 0.2556 | |
| 10 | 0.0639 | 0.639 | |
| 20 | 0.1278 | 1.278 | |
| 62.597 | 0.40 | 4.00 | |
| 78.247 | 0.50 | 5.00 | |
| 200 | | | |
| 300 | 1.917 | 19.17 | |
| 1500 | 9.583 | 95.83 | |
| 2000 | 12.777 | 127.77 | |
| 3000 | 19.166 | 191.66 | |
| 6000 | 38.332 | 383.32 | |

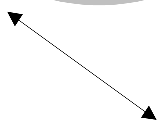
# 14. Context Diagram Conventions



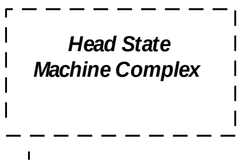

**a Finite State Machine (FSM)** which handles a specific task. Each of which "capable" of communicating with any other FSM in the systems. FSM can use any of the sevices provided by the OS Kernel.



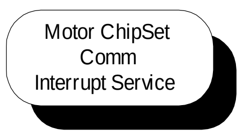

**a Coordinator.** A Finite State Machine in itself whose task is to handle or manage a set of subordinate state machines.



**a Communication direction line.** In a Context Diagram this line will server as a guide as to determine which FSM is allowed to communicate to another. Although any FSM is capable of communicating with any other FSM in the systems, Communication Flow should not be broken.

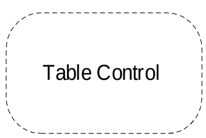



**a State Machine Complex** comprises of a manager FSM and its subordinates. The label allows special indication for such a dedicated system of FSMs.

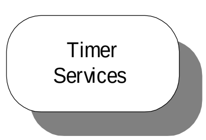



**a (Low Level) Driver** to indicate the existence of an Interrupt Servicing Routine. Unlike other components of the system, these drivers are initimately coupled to the Hardware Perhiperals as well as the RTOS Kernel.

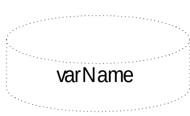

**a Data Manager** providing access to a component of the system. This accomplishes the "data hiding" aspect of an object oriented system.




**a Virtual Interface** providing control over a system component.




**a Real Time OS service.** This represents a service or group of services that the Kernel provides the system.

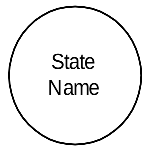



**a Data Store.** This represents a storage in memory. A variable in short. Can also be a structure or arrays...Connection to the data store indicates how the data is shared among the different objects.

HCC:EnableSComm

**a Message "EnableSComm" sent by HCC** .

# 15. State Transition Diagram Conventions




**a State Bubble**

A circle with a State Name in the center. This represents the state of a given machine.

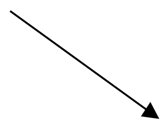
GotoXYTarget:MTRC

**a Message "GotoXYTarget" sent to MTRC** .

( x== 10)

**a Decision Statement** .

a State transition is sometimes is determined by a Decision Statement. This symbol shows the reason why the state changes from one to another.



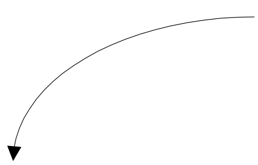
**an Incoming Communication line.**

This is a heavy line with an arrow pointing towards the center of a State Bubble. This line is always accompanied with a text indicating the message and the sender ID

**SMNAME_h1**

**an Exit Procedure**

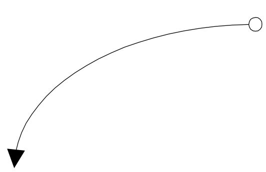
Also known as a Message Handler.
Each state transition begins by excuting an Exit Procedure which performs all of the necessary things required to change the State Machine's state. SMNAME_h1 is the actual label of the exit function as written in the source code.



**a State Transition Path Line**

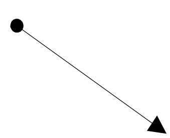
This is a light curved line that begins from one state bubble and ends at another state bubble. This represents the how the states will chang e bas e on the current event or system conditions.

Note: If the state stays the same then this line will point back to the same state bubble.



**a State Transition Branch Path Line**

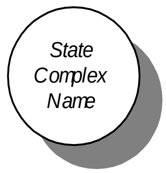
This is the same as a State Transition Path Line except that it begins with a light dot. This Line symbol is used when the state can change to more than one state. This line is usually accompanied with a decision statement.



**an Out-Going Communication line.**

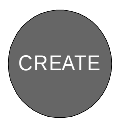
This is a light line that begins with a heavy dot and ends with with an arrow. The arrow points away from a State Transition Path Line. This Communication line is always accompanied with a text indicating the message and the receiver ID

# 16. State Transition Diagram Conventions (continued)




**a State Complex Bubble**

A circle with a State Complex Name (in italics) in the center. This represents a group of state transitions. Use this to make a complex group of state transition abstract. The state transitions represented by this diagram is further elaborated on the next level.




**the CREATE Bubble**

A dark circle with the text "create" at the center. This represents the initial state of the machines when power is first applied.


**an Italicized Exit Procedure**

This indicates that the exit procedure is reused in another state.