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 8:57:00 AM
Date Printed 12/18/2017 8:57:00 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 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

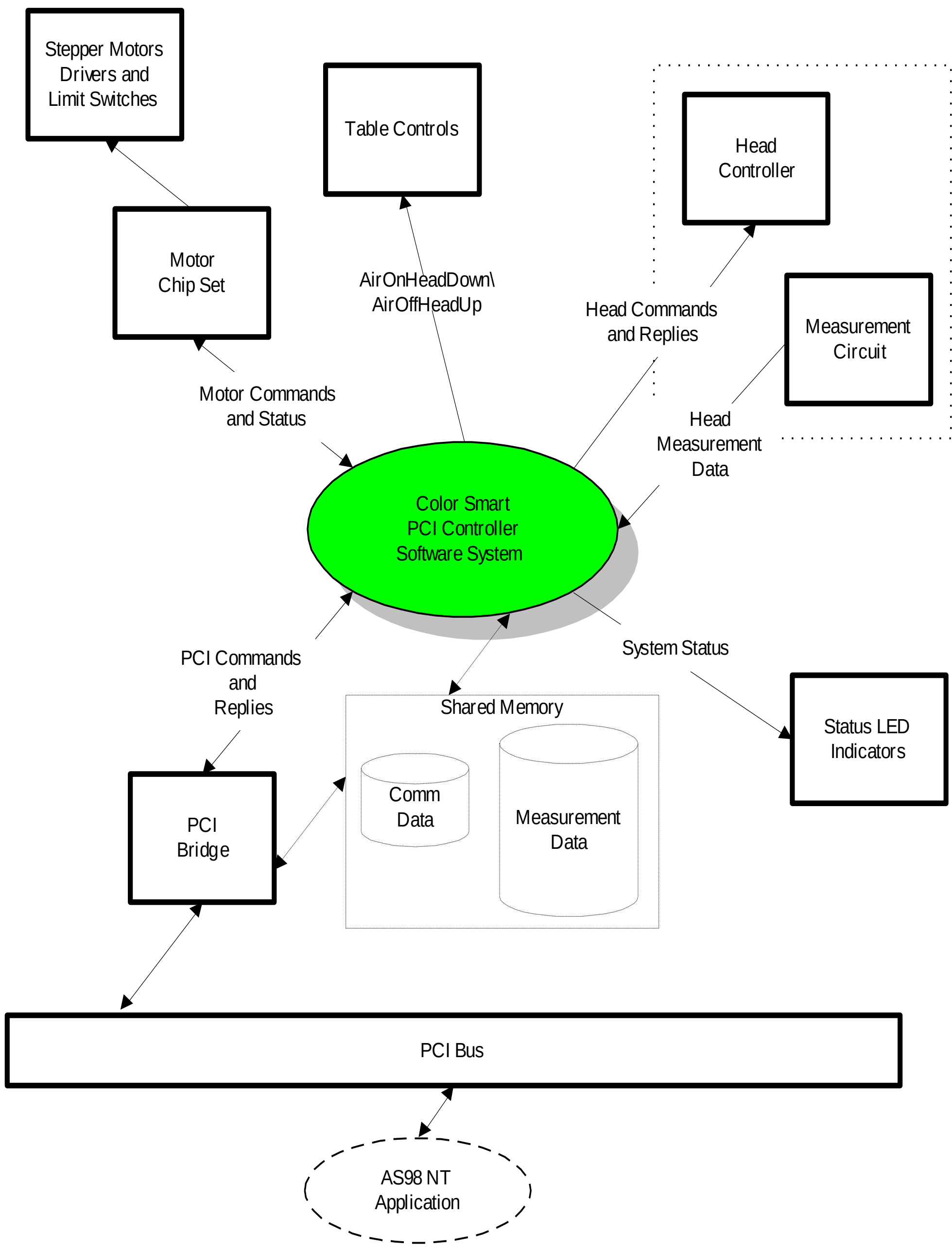
WHAT'S NEW (since 00.15)

- ✓ Move Profile DM has been replaced with Table Parameters Data Manager. Move profiles are within this manager. All references to TPDM has been updated.
- ✓ HMM Do Color Bar Scan was state transitions were incorrect, now revised and corrected.
- ✓ MTRC, added many descriptions and details all throughout, errors corrected – 85 % complete.
- ✓ MTRC added GotoXTarget and GotoYTarget state transitions (for one axis moves)
- ✓ CSM task definition updated
- ✓ CSM Initial State Transition revised
- ✓ XAT and YAT, added details, errors corrected
- ✓ Inputs from all reviewers (rev 00.15) are implemented
- ✓ Moved MDM section close to MCM section (measurement data stuff)
- ✓ Reorganized HMM section sequence
- ✓ HMM, added many descriptions and details all throughout, errors corrected – 85 % complete.
- ✓ Added IMPORTANT notes in RED
- ✓ HMM added a new service "Measure At Point"
- ✓ The color bar scan is more defined
- ✓ Created a Boot Kernel context diagram

1. Table of Contents

2. High Level Hardware to Software Interface Diagram

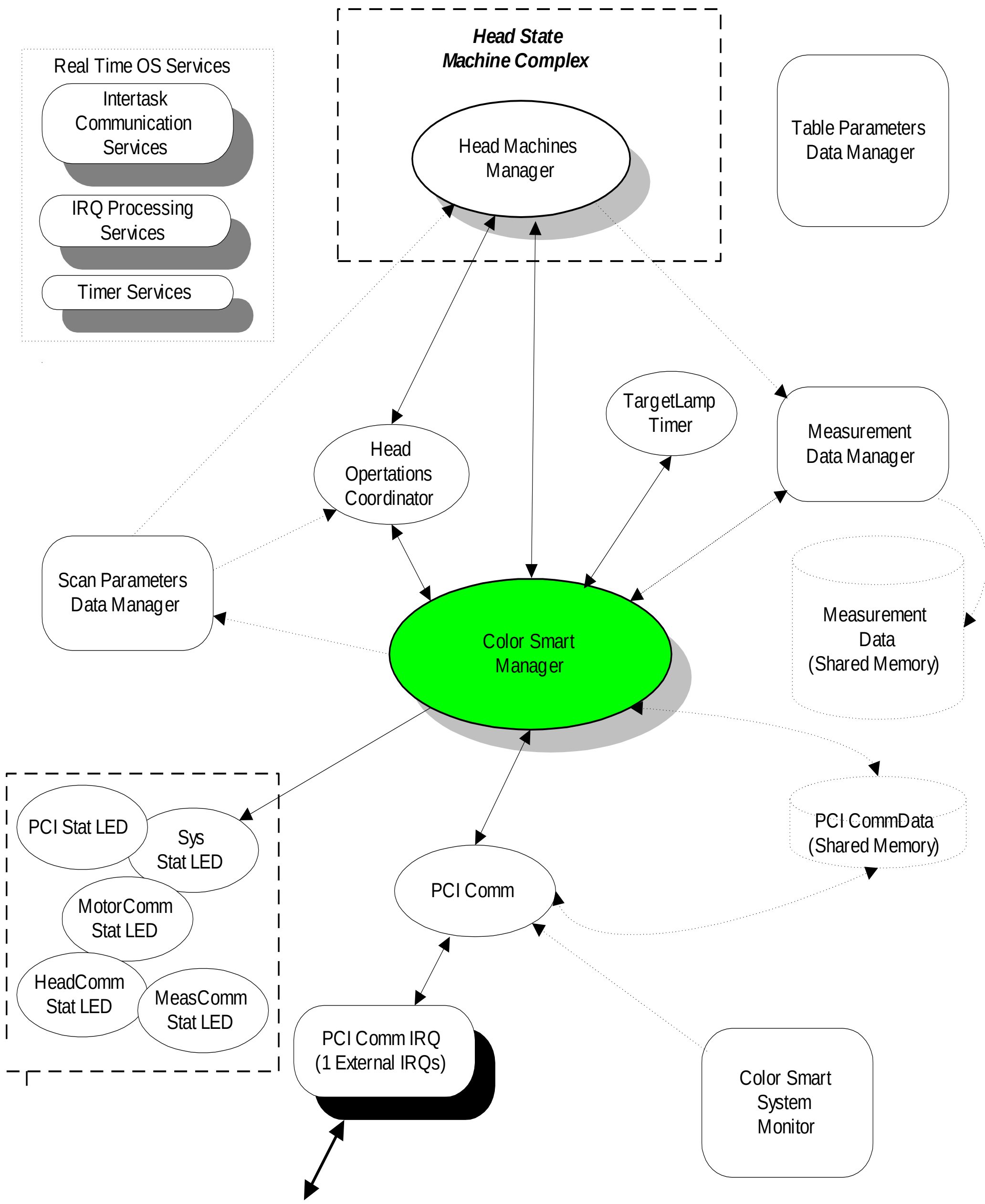
The diagram below shows the relationship of the Color Smart PCI software to the hardware environment providing us an overview of how the Color Smart PCI Software interfaces to the outside world:



3. High Level Software Context Diagram

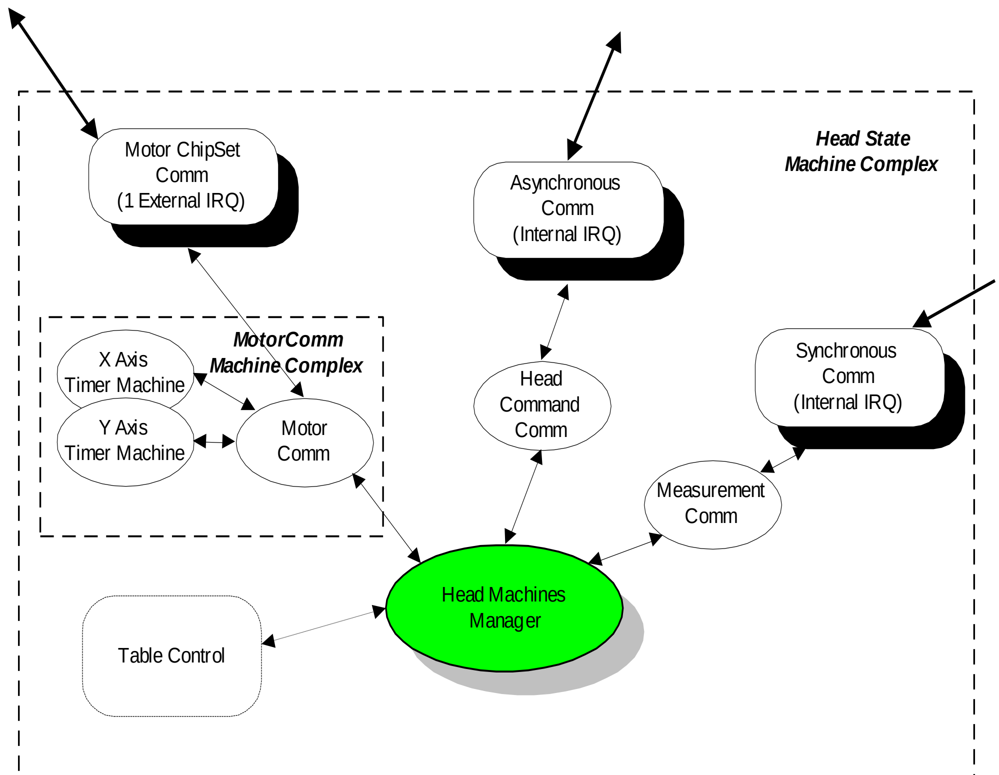
The diagram below shows the different software components involved in the operation of the Color Smart PCI system. Some of these components is an interface to the hardware components as seen in the previous diagram.

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



3.1. 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 Software Interface Diagram.



4. Color Smart System Overview

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.

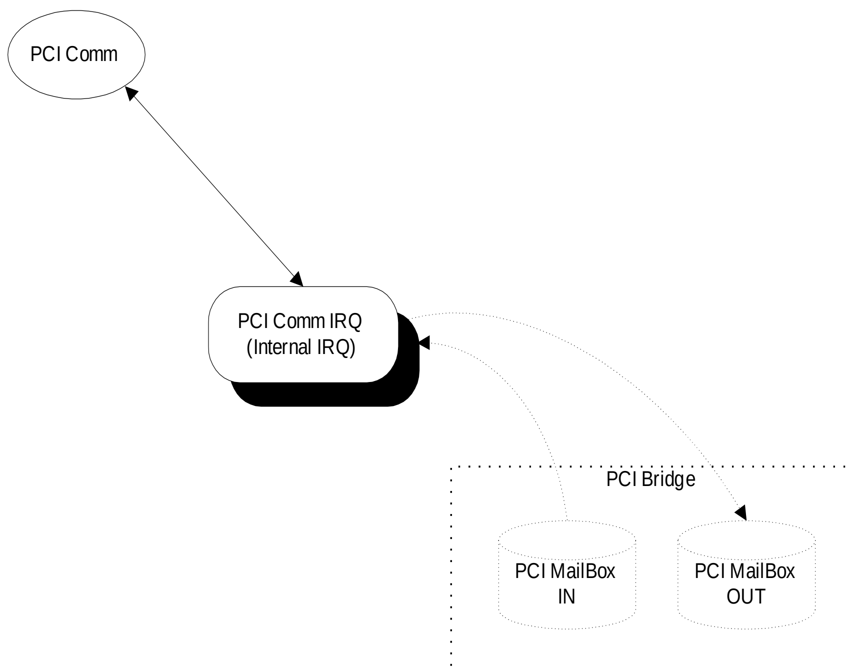
5. Task Descriptions

The Task Description sections describe **what** each tasks will need to accomplish. These sections do **not** talk about **how** they will be accomplished.

5.1. 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.

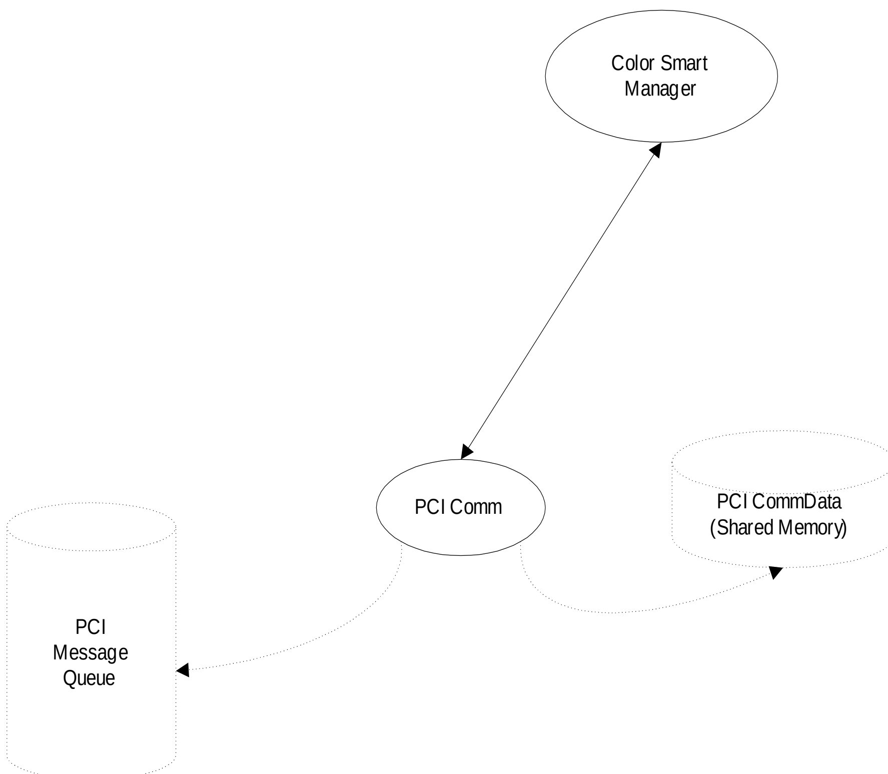


5.2. PCI Comm (PCM) Task Description

PCM - P*ci *C*om*M

This state machine (SM) will handle all the communication that is transacted between Color Smart NT Application 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 Color Smart NT Application will be able to communicate with the Color Smart PCI at any point of the Color Smart PCI operation.

Due to the PCI Communication Protocol requirements, 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.



5.3. 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:

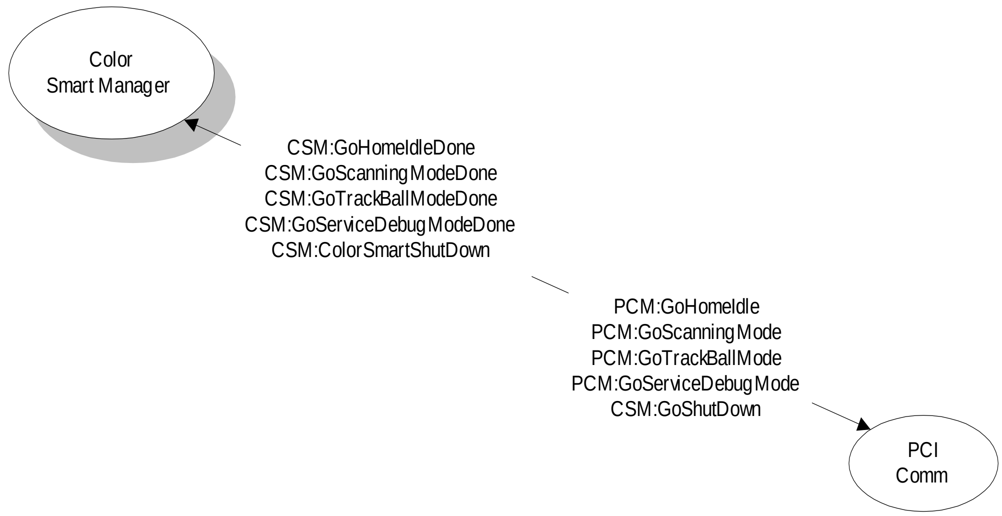
- He knows the overall condition of the system
- He knows the command set specification
- He is responsible for generating the replies to the commands given by the Color Smart NT App
- He decides when a request can be accepted or denied
- He makes sure that the message data is within their valid ranges 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
- He is in charge decides which move profiles to use and instructs HMM about the move profile changes as necessary
- In general, the CSM states are representative of what the system as a whole is currently doing
- He is the boss.

5.3.1. 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. **SERVICE DIAGNOSTIC Mode**

5.3.1.1. CSM – System Interaction, *Mode Change Requests*

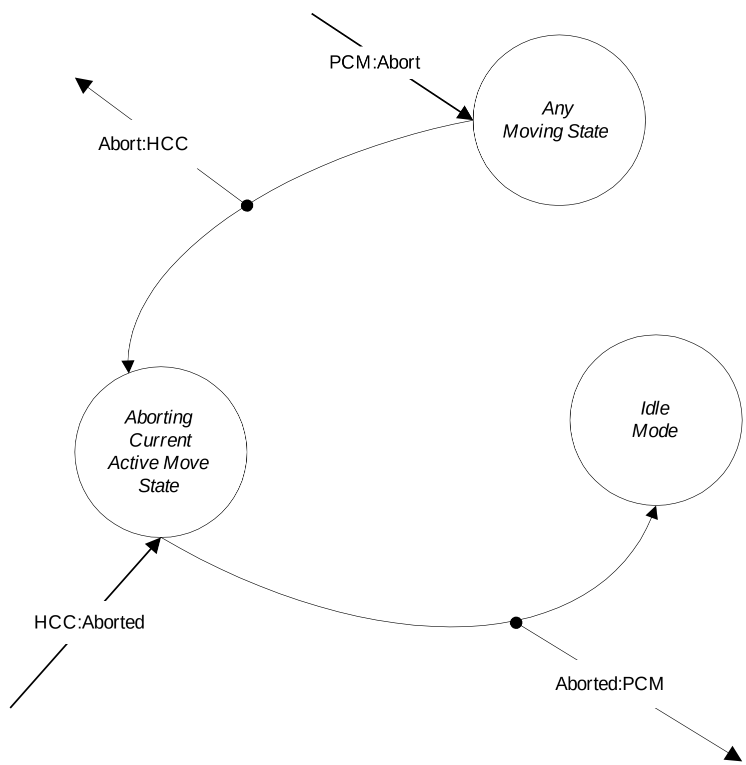


5.3.1.1.1. CSM – Abort, State Transitions

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. Service 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.



5.3.1.1.2. 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.

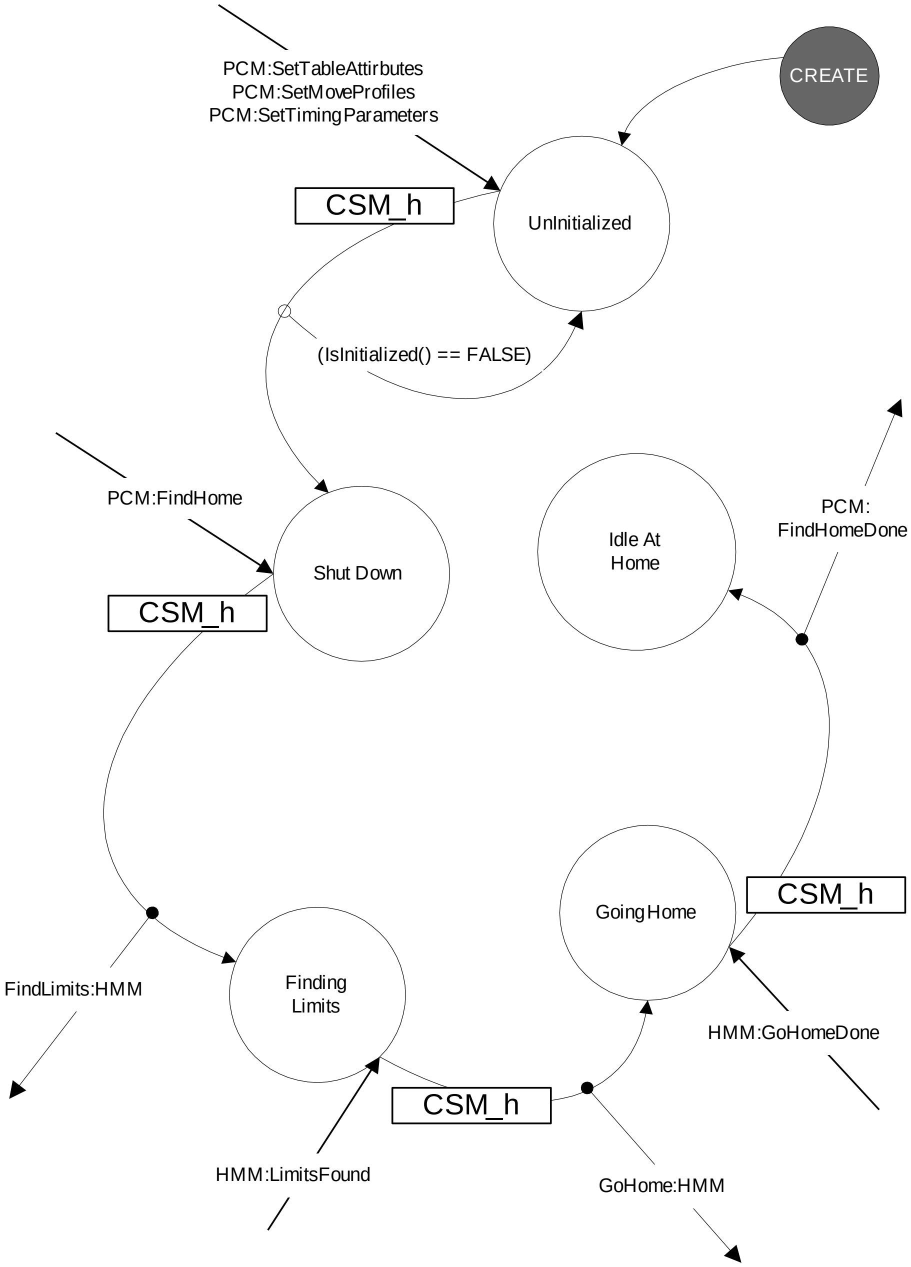
Once CSM has moved from the UN-initialized State into the “Shut Down” state, a Find Home command needs to be given by the NT Host before CSM can proceed to the Idle At Home State and later to other states for normal move operations.

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. SERVICE DIAGNOSTIC Mode

The next diagram shows all of these initial state transitions that need to take place.

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.

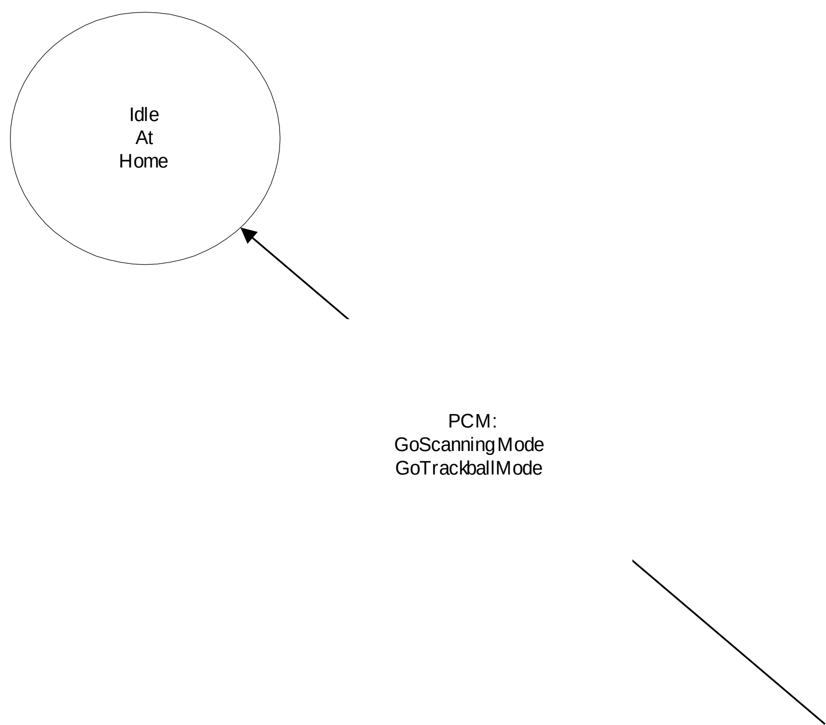


Note:

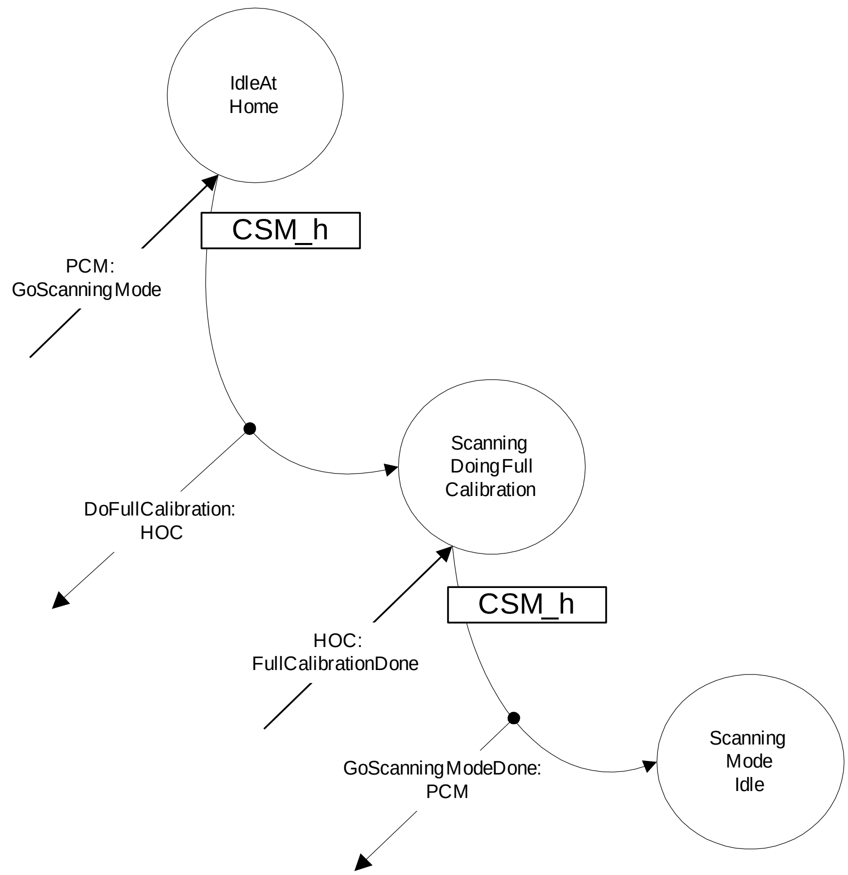
When CSM is in the Home Idle State, the air is off and the measurement sensor is at the middle of the white plaque. This is considered the “parked” position.

5.3.1.1.3. CSM – Idle At Home, Messages Handled

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.

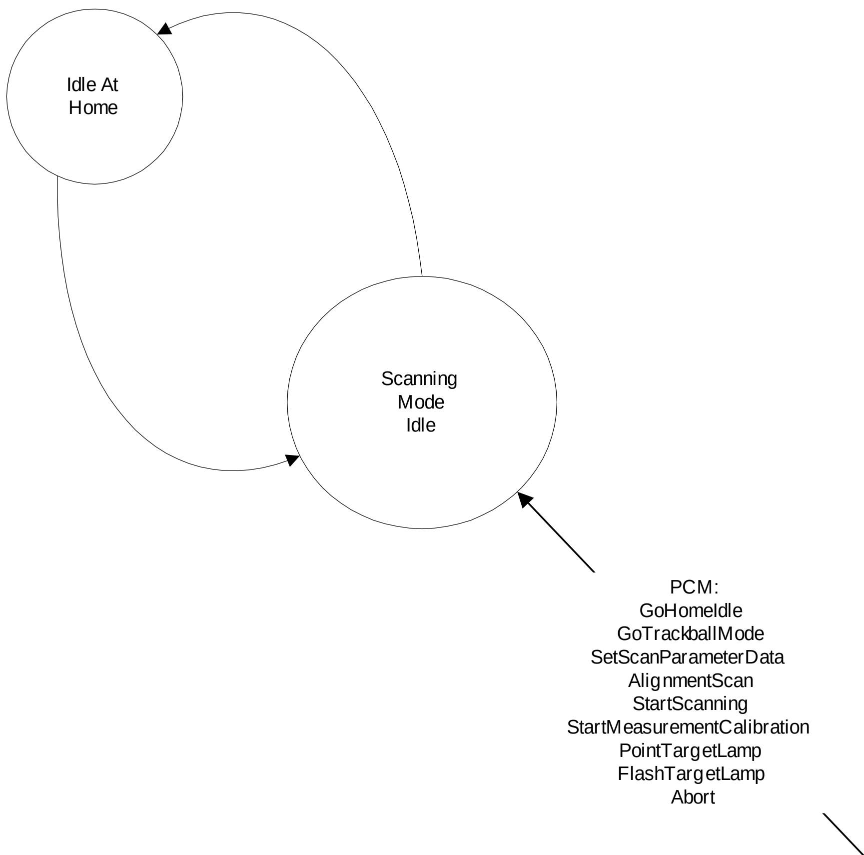


5.3.1.1.4. CSM (Idle At Home) – Go Scanning Mode, State Transitions

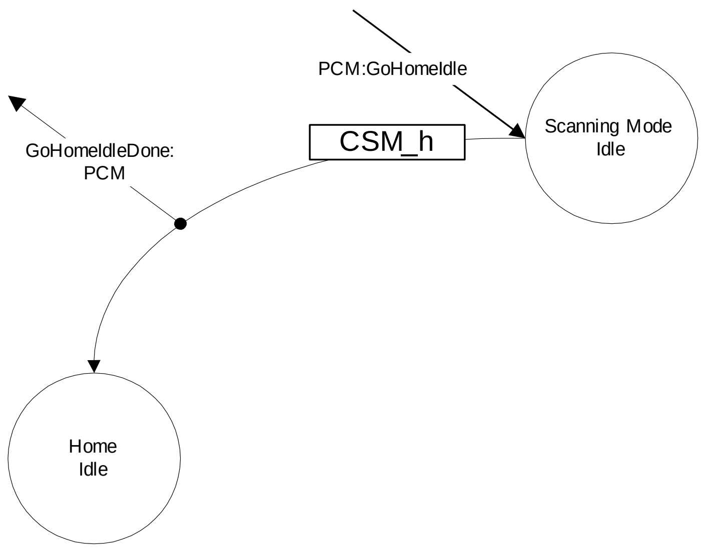


5.3.1.1.5. CSM – Scanning Mode Idle, Messages Handled

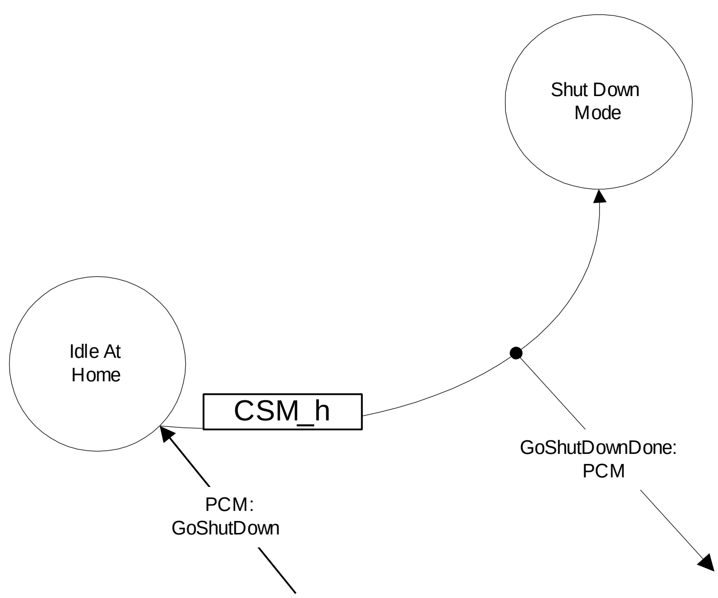
The diagram below shows all the messages that CSM will handle while it is in the Scanning Mode Idle State. The next pages will show how CSM handles each of these messages as well as the state transitions that occur based on the event.



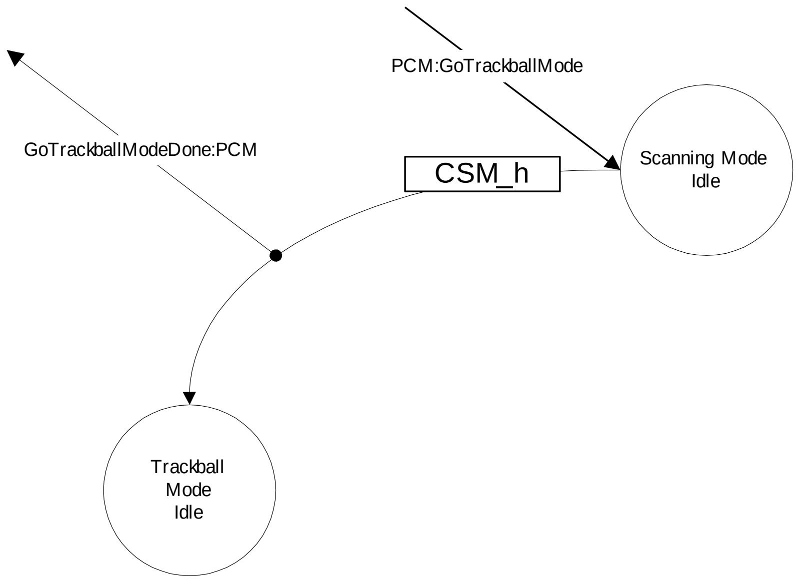
5.3.1.1.6. CSM – (Scanning Mode Idle), GoHomeIdle, State Transitions



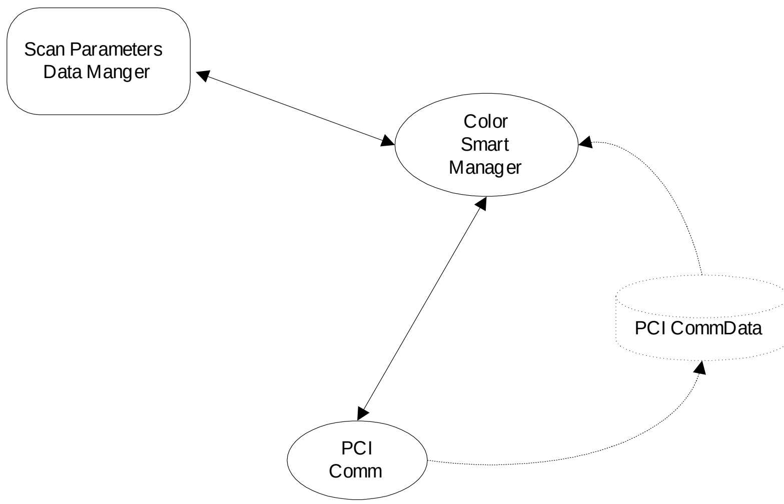
5.3.1.1.7. CSM (Idle At Home) – Go Shut Down, State Transitions



5.3.1.1.8. CSM – (Scanning Mode Idle), GoTrackballMode, State Transitions

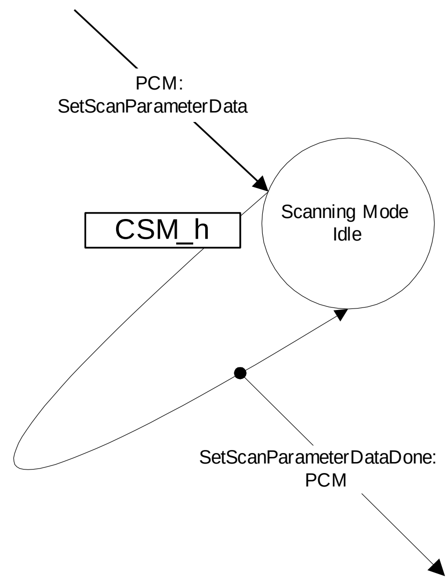


5.3.1.1.9. CSM – System Interaction, Scanning Mode Idle, Set Scan Parameter Data

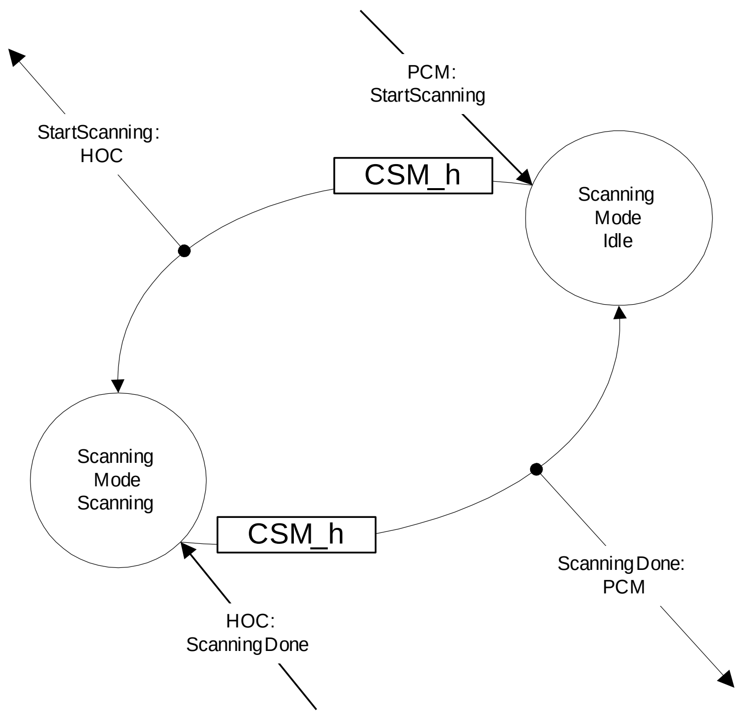


5.3.1.1.10. CSM – (Scanning Mode Idle), Set Scan Parameter Data, State Transitions

This message tells CSM that the scan parameter data is available. CSM gives the scan parameter data that is in the PCI Comm buffer over to the Scan Parameter Data Manager (SPDM).



5.3.1.1.11. CSM (Scanning Mode Idle) – Start Scanning, State Transitions



5.3.1.1.12. CSM – System Interaction, Scanning Mode Idle, Alignment Scan

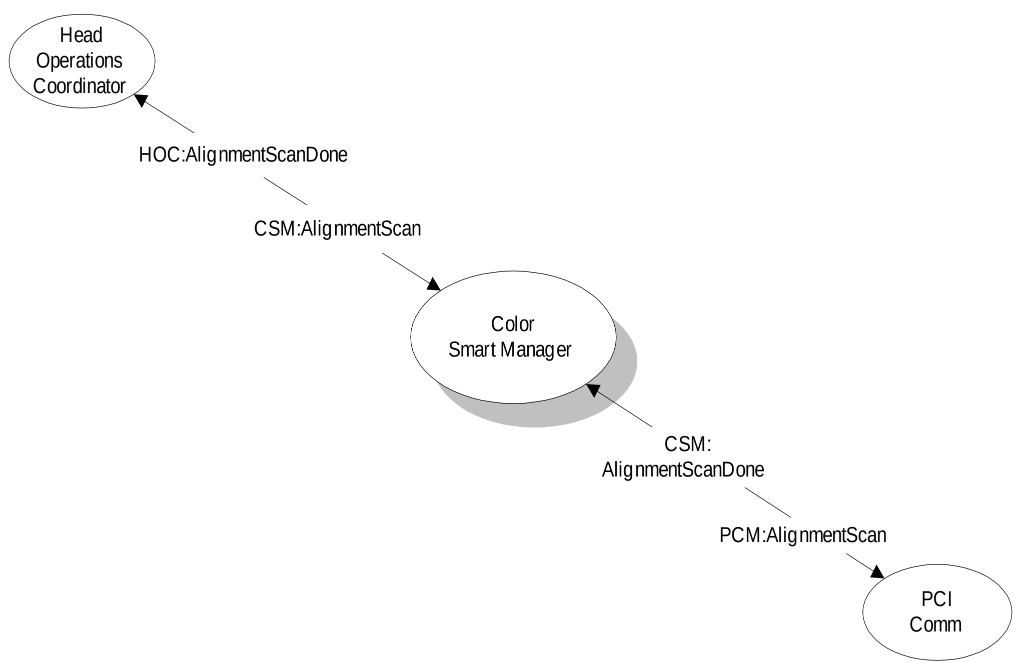
IMPORTANT:

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

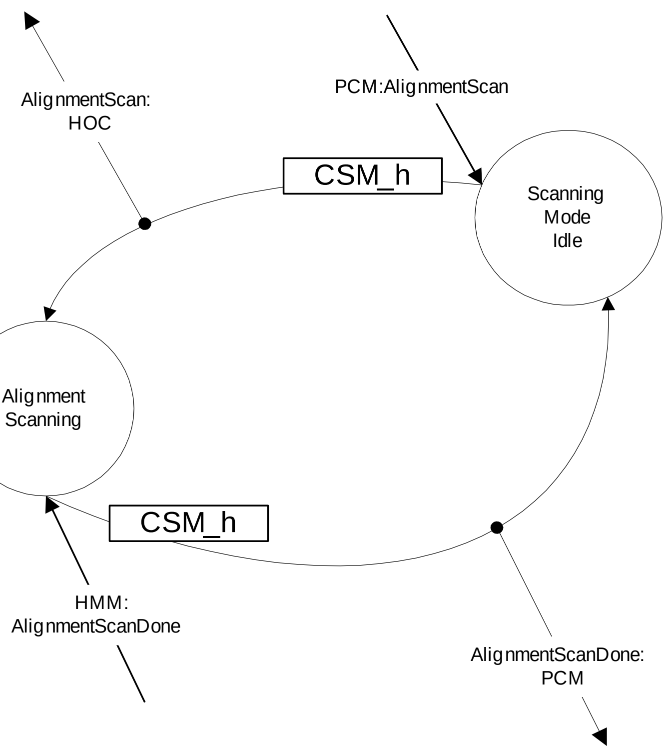
X1 = 333, Y1 = 200
X2 = 777, Y2 = 200

CSM will reject the message if this rule is not followed.



5.3.1.1.13. CSM – (Scanning Mode Idle), Alignment Scan, State Transitions

The Alignment Scan message is accompanied with data that is stored in the PCI Communication Buffers.

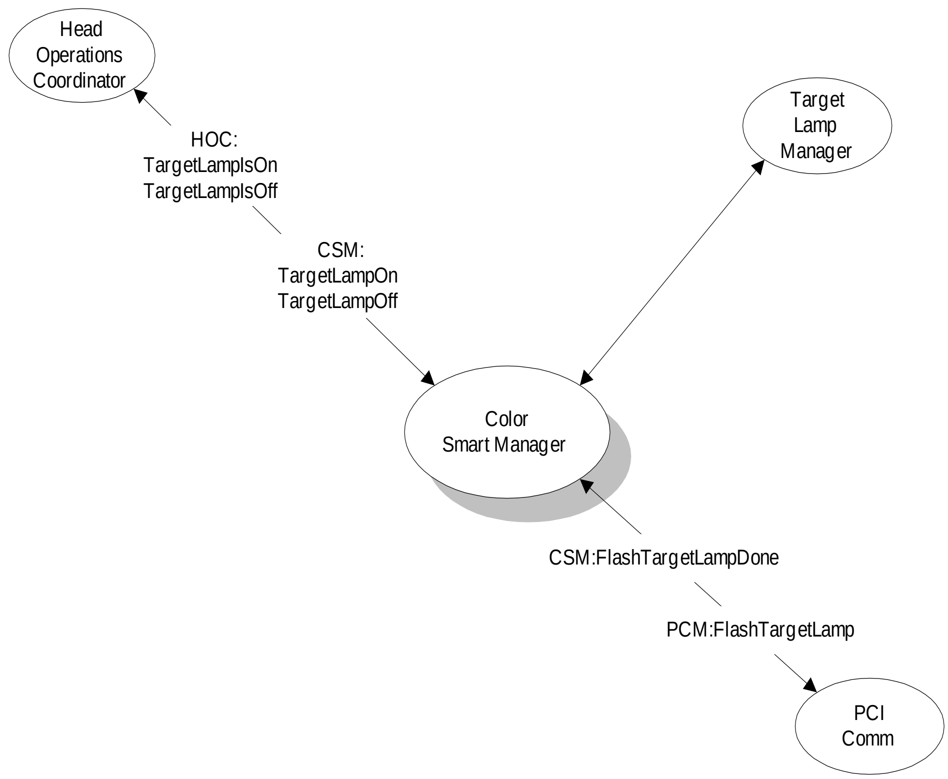


Note:

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.

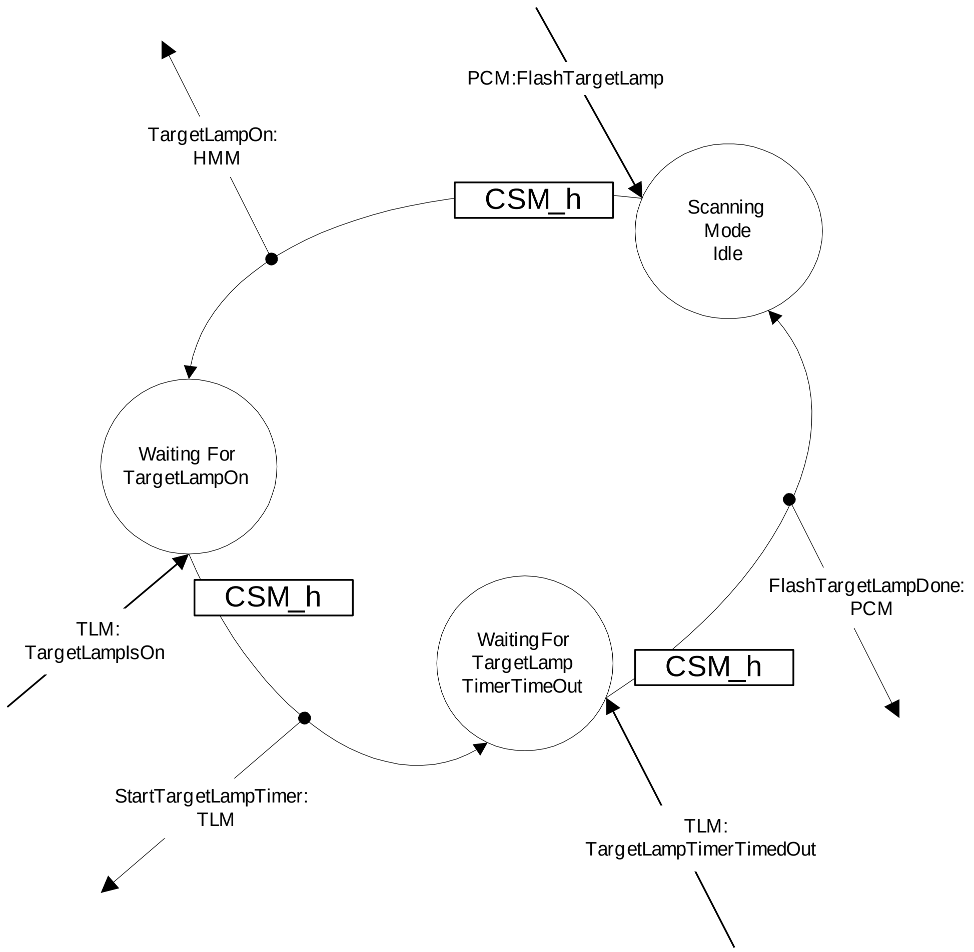
Before the Alignment Scan procedures are executed, the NT Application may perform a "Classify Alignment Bar" procedure. This procedure allows the NT Application to determine the orientation of the alignment bar. To perform the Classify procedure, the NT Application will simply send a set of point to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

5.3.1.1.1. CSM – System Interaction, Scanning Mode, Flash Target Lamp

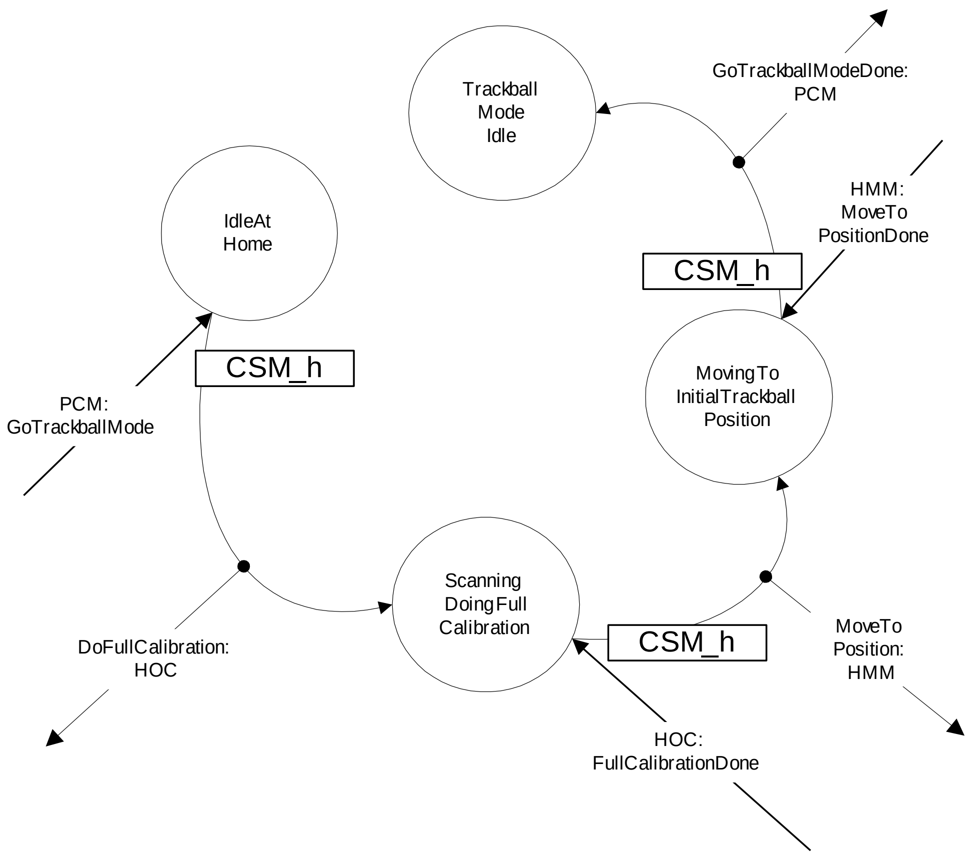


5.3.1.1.1. CSM – (Scanning Mode Idle), Flash Target Lamp, State Transitions

D1 will contain how long the target lamp stays on (in 100 msec units).

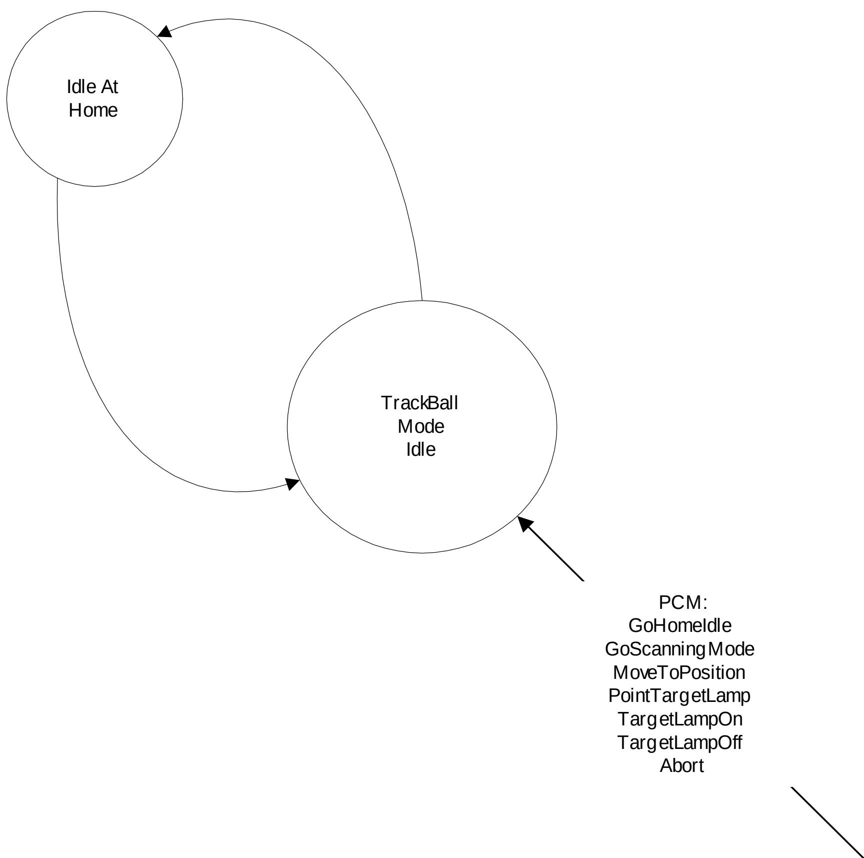


5.3.1.1.2. CSM (Idle At Home) – Go Track Ball Mode, State Transitions



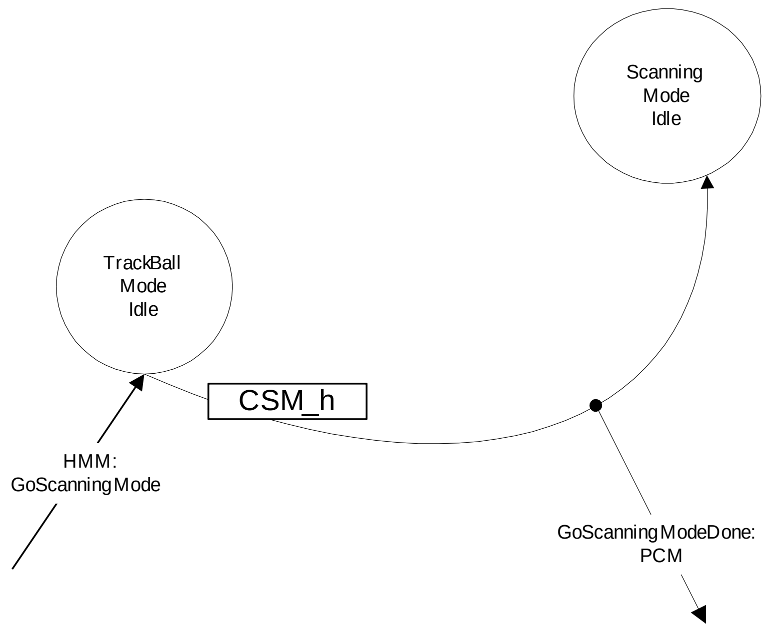
5.3.1.1.3. CSM – Track Ball Mode Idle, Messages Handled

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.

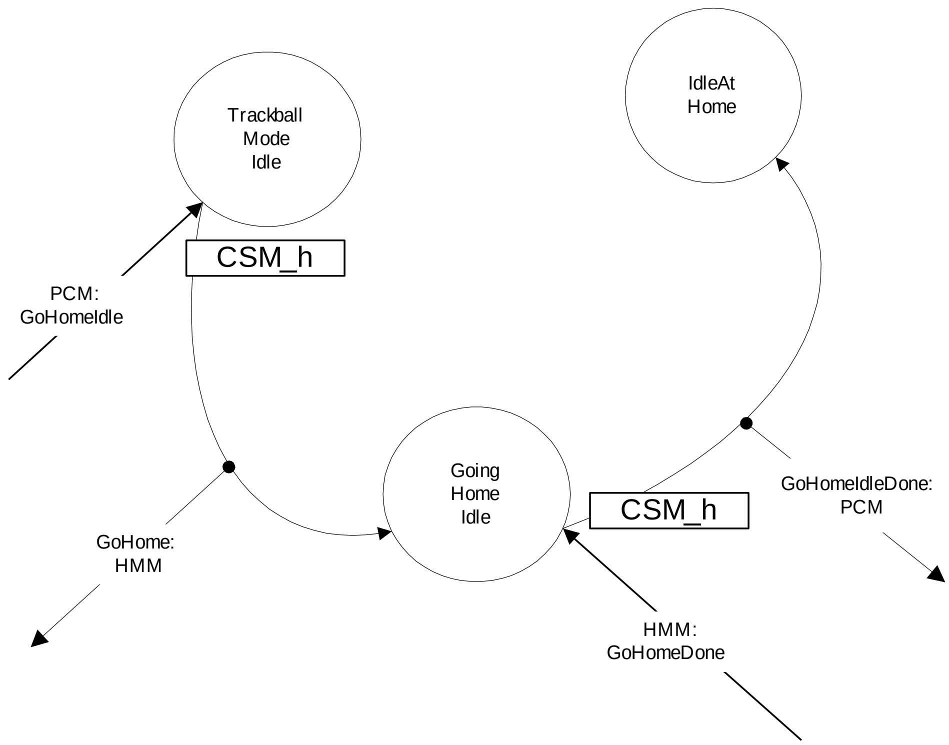


5.3.1.1.4. CSM – (Track Ball Mode Idle), Go Scanning Mode, State Transitions

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 will be executed once the mode change has been made.



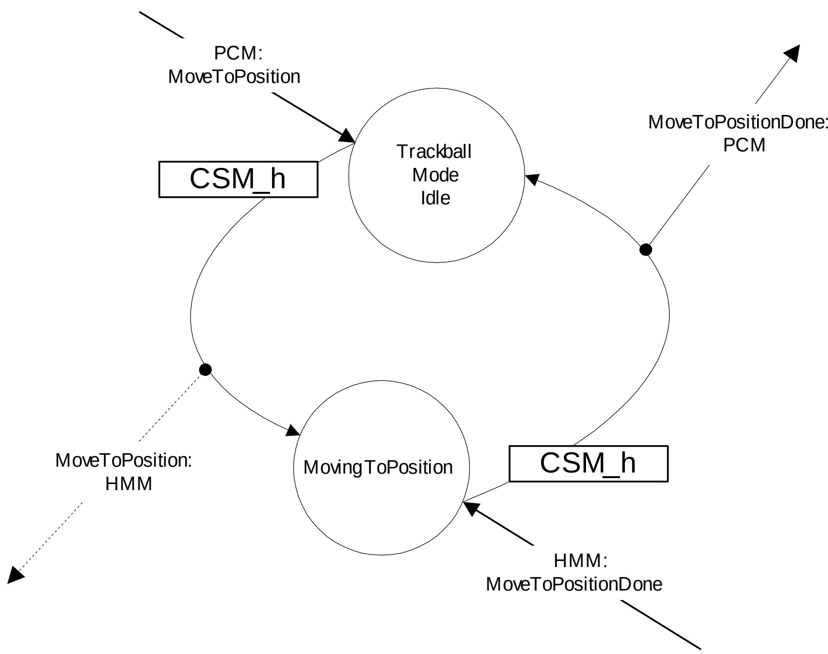
5.3.1.1.5. CSM (Track Ball Mode Idle) – Go Home Idle, State Transitions



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

Sending this message performs Trackball moves. The NT Application is responsible for making all the calculations so that the moves will be smooth. The NT Application will also take into consideration the acceleration of the trackball. This acceleration “profile” 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.

PCM sends the MoveToPosition message with data. D1 has the X coordinate and D2 has the Y coordinate. CSM then relays the message to HMM. Once the move is complete, HMM will send a message about this event. This message is then finally relayed back to PCM.

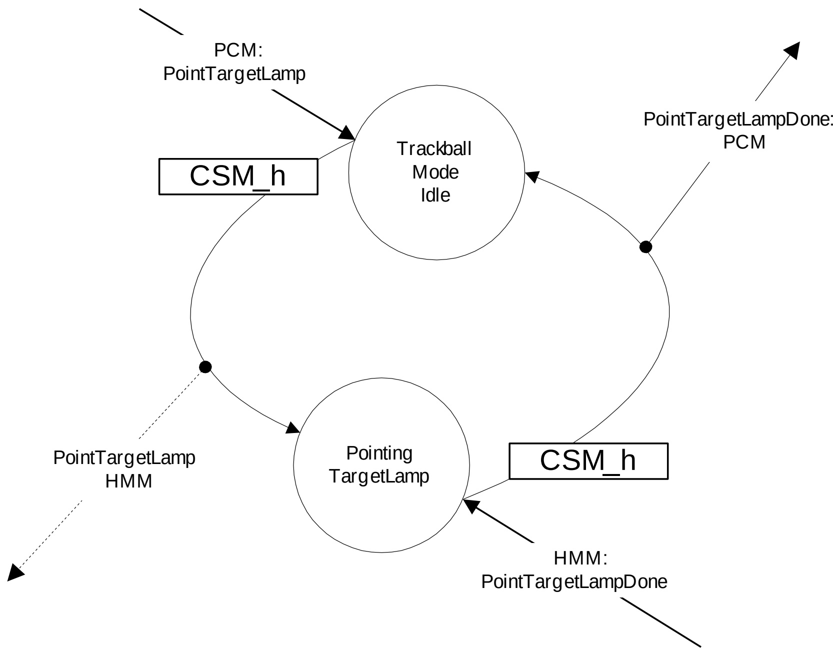


Notes:

- 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

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

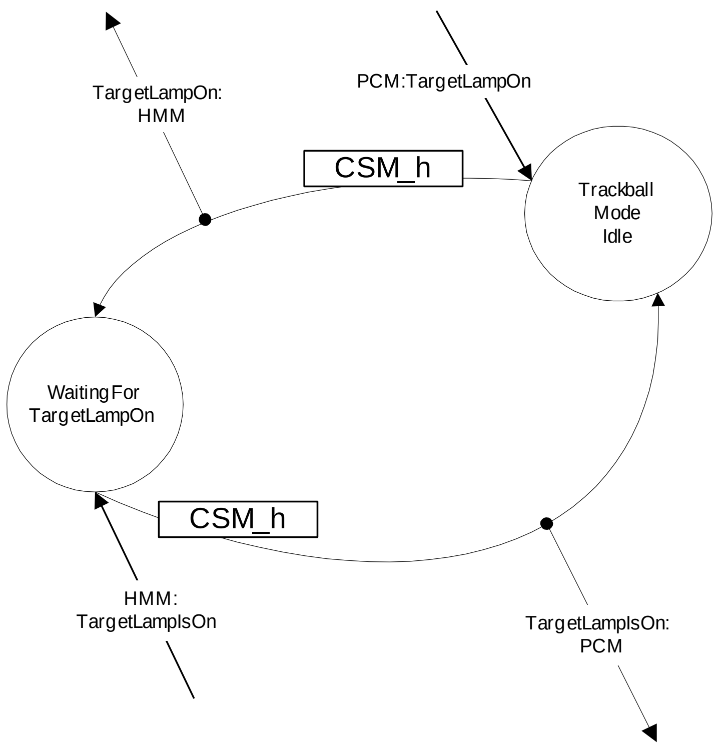
This request is similar to the “Move To Position” message except that the position requested will be the center of the Target Lamp instead of the measurement sensor.



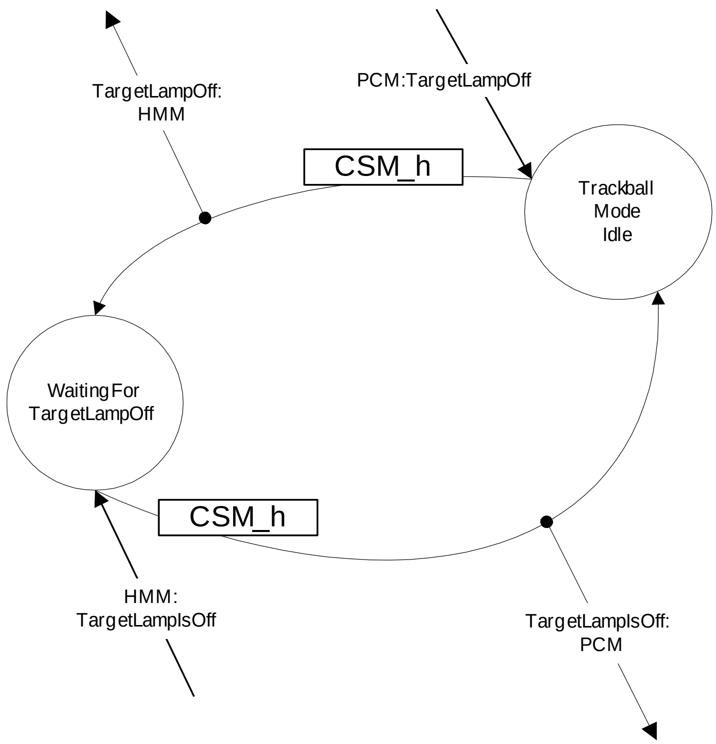
5.3.1.1.8. CSM – (Trackball Mode Idle), Target Lamp On, State Transitions

Note:

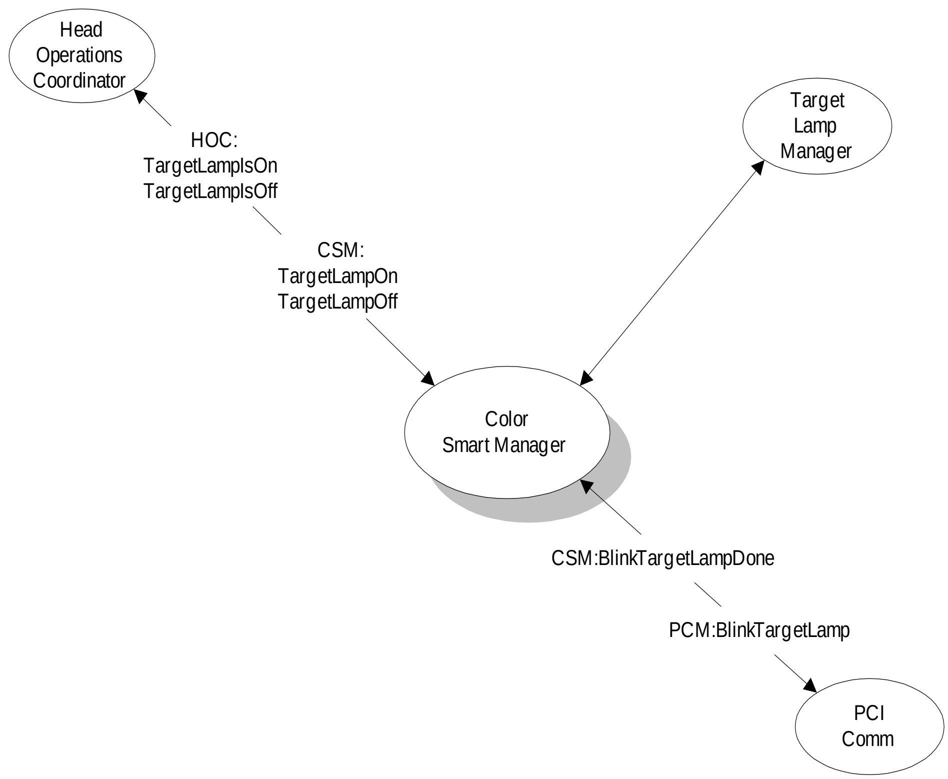
The controller board will automatically turn off the target lamp after some time of trackball inactivity. It will automatically be turned on once activity resumes.



5.3.1.1.9. CSM – (Trackball Mode Idle), Target Lamp Off, State Transitions

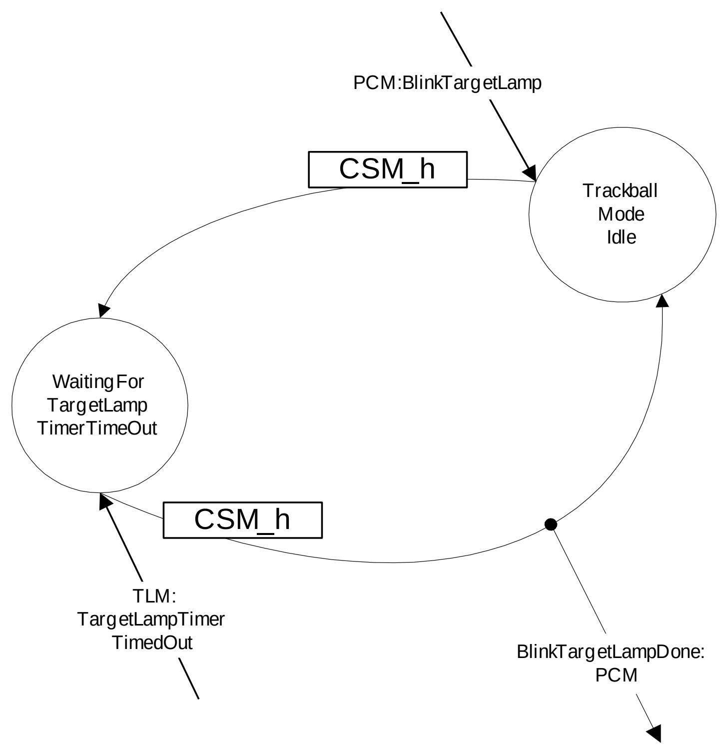


5.3.1.1.10. CSM – System Interaction, Trackball Mode, Blink Target Lamp

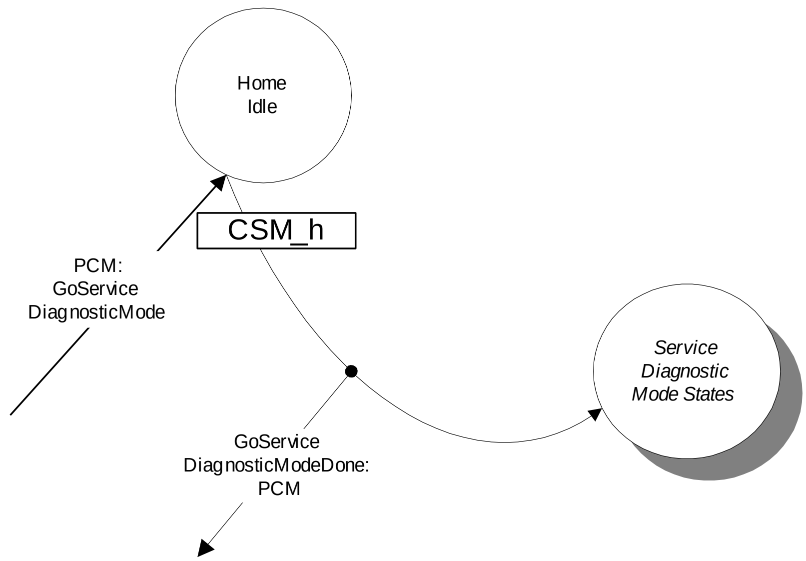


5.3.1.1.11. CSM – (Trackball Mode Idle), Blink Target Lamp, State Transitions

D1 will contain how long the lamp stays off (in 100 msec units).

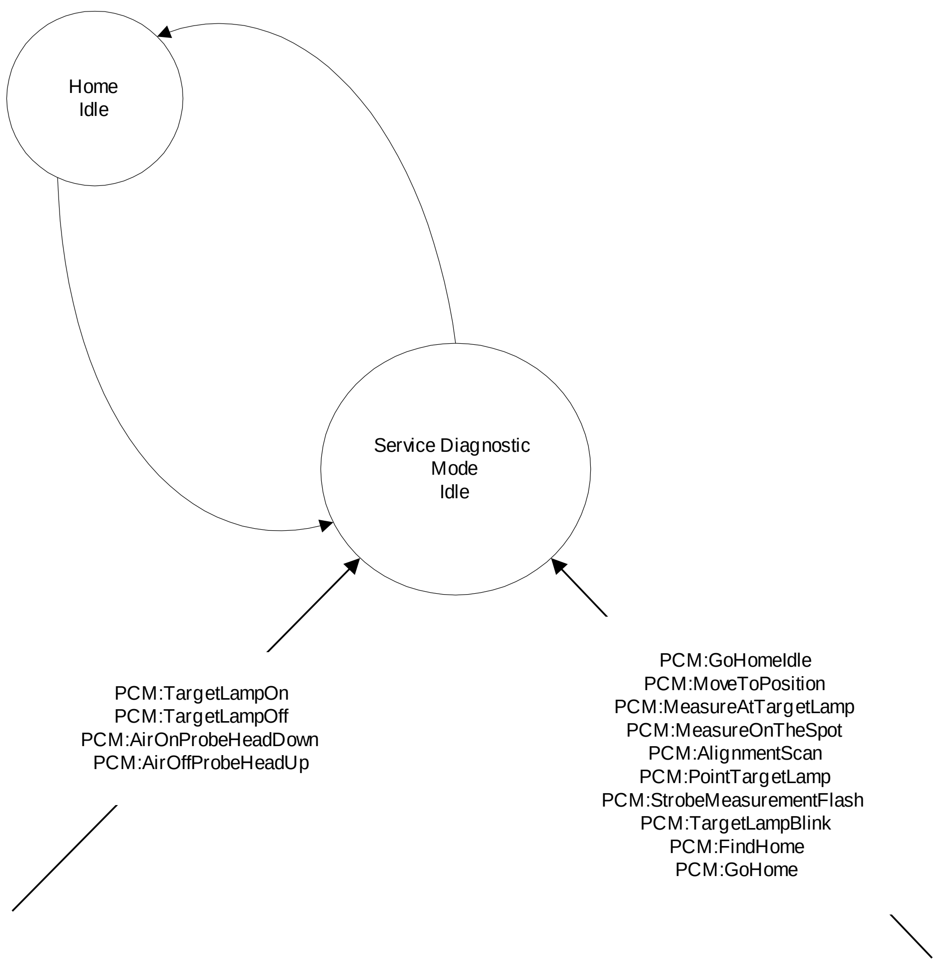


5.3.1.1.12. CSM – Go Service Diagnostic Mode, State Transitions

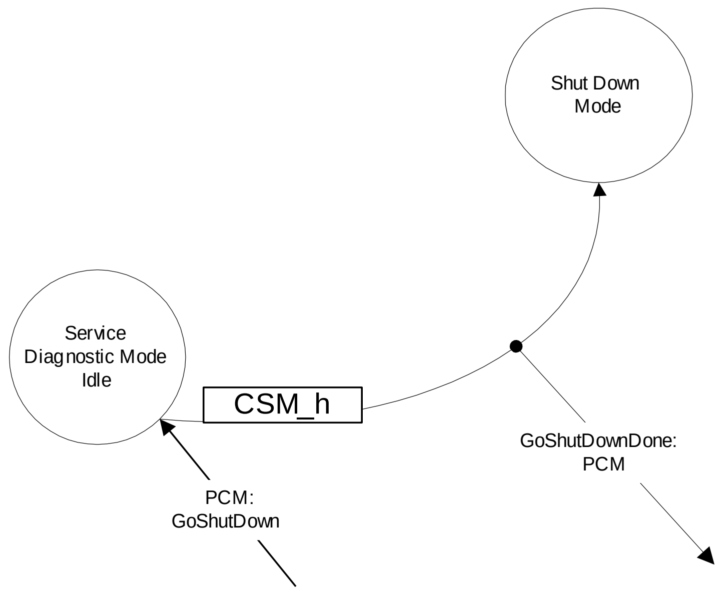


5.3.1.1.13. Service Diagnostic Mode Idle, Messages Handled

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.

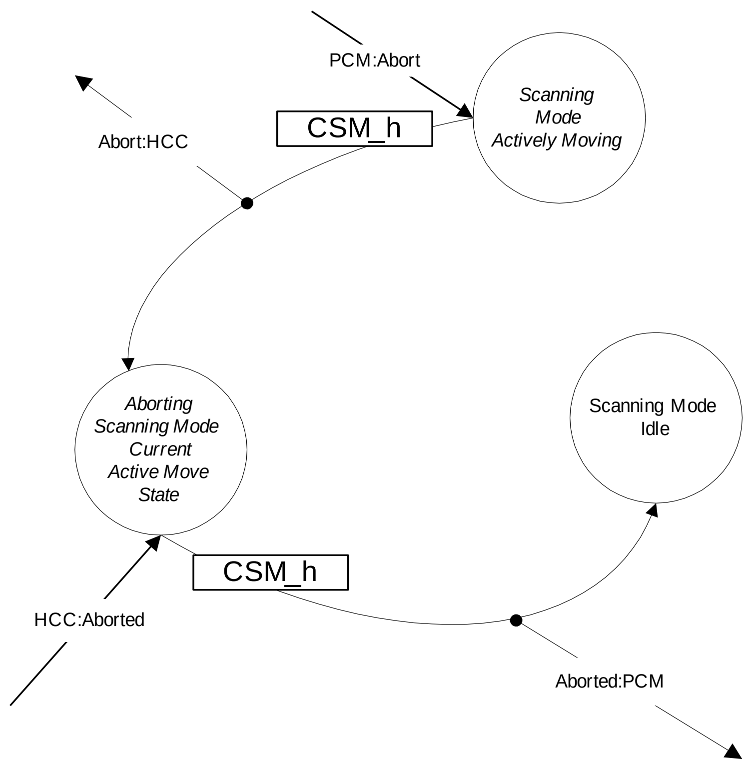


5.3.1.1.14. CSM (Service Diagnostic Mode Idle) – Go Shut Down, State Transitions



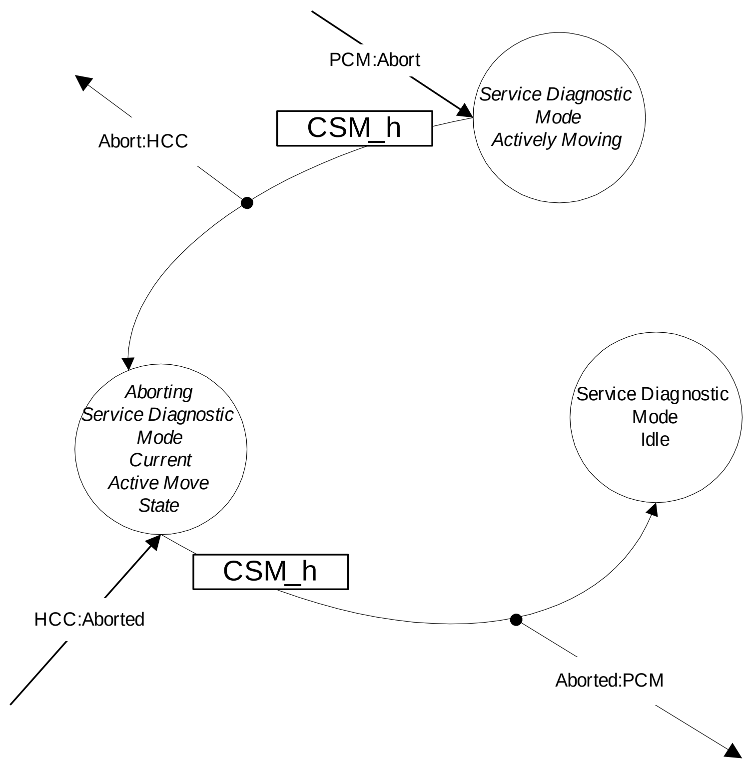
5.3.1.1.15. CSM – (Scanning Mode Actively Moving), Abort, State Transitions

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.

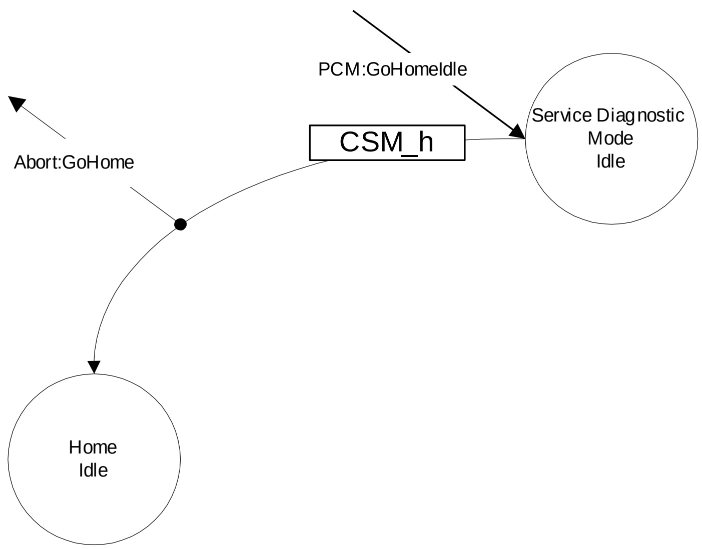


5.3.1.1.16. CSM – Service Diagnostic Mode, Abort, State Transitions

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.

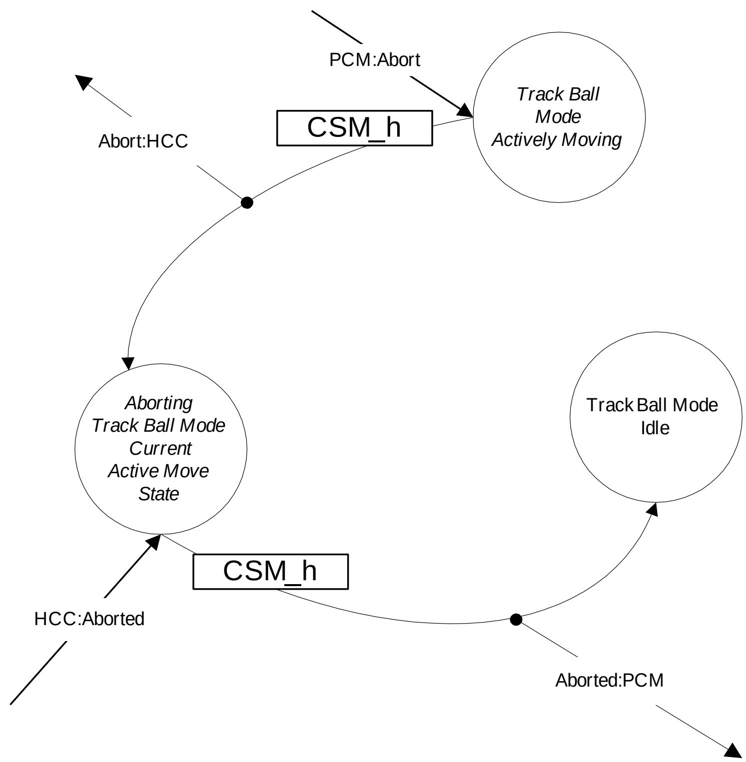


5.3.1.1.17. CSM – Service Diagnostic Mode, GoHomeIdle, State Transitions



5.3.1.1.18. 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.



5.4. Synchronous Comm IRQ (ISCM) Task Description

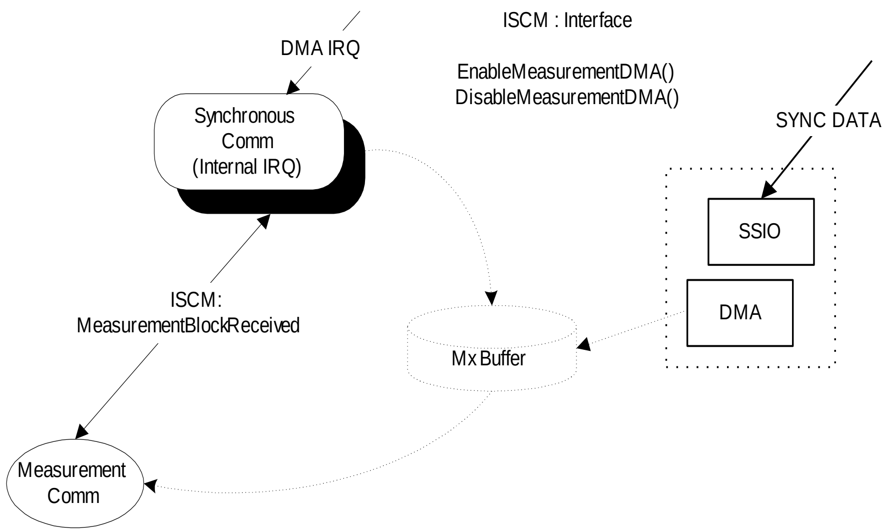
ISCM - I*nterrupt* ***S****ynchronous* ***C****om****M***

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 stuff the SSIO (Sync Serial I O) data into a buffer. Once the buffer is full, the DMA issues an interrupt (MxBufferFull). Through the Kernel, MCM receives the message notifying him about the event.

MCM then takes the Measurement Data to perform further processing (if needed).

5.4.1. 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.

5.5. Measurement Comm (MCM) Task Description

MCM - Measurement ComM

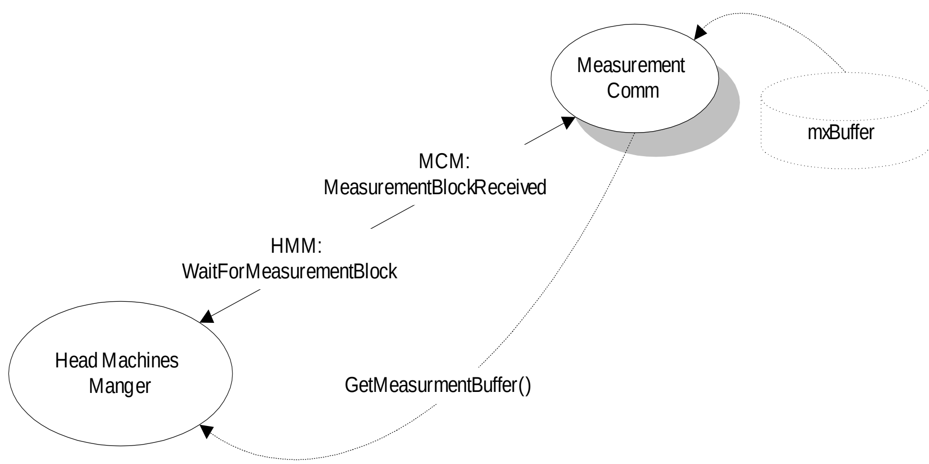
This will handle all the measurement data transferred between the Head Scanner and the Color Smart PCI Card. The state machine will handle the task based on a Communication Protocol. The task will include Timeout detection where the timing services of the OS Kernel will be utilized.

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.

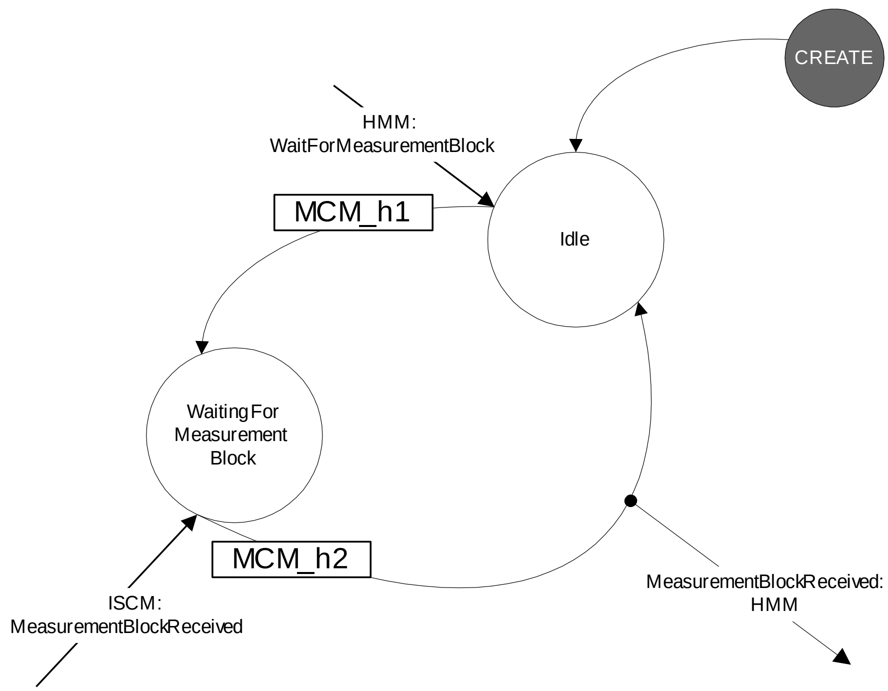
5.5.1.1. MCM – System Interaction, WaitForMeasurementBlock



Note:

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

5.5.1.1.1. MCM – WaitForMeasurementBlock, State Transitions



5.6. Measurement Data Manager (MDM) Task Description

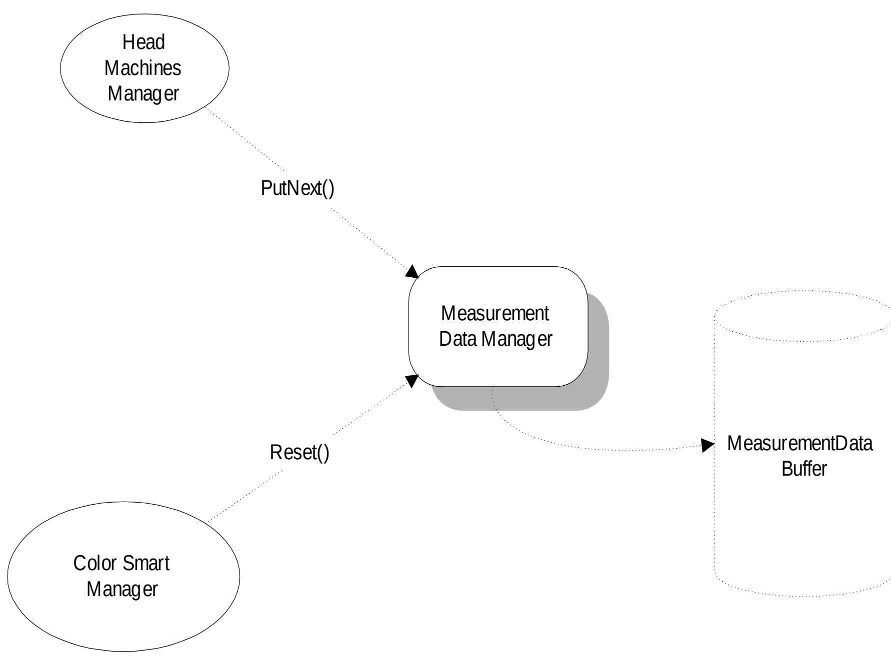
***MDM** - **M**easurement **D**ata **M**anager*

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.

5.6.1. MDM – High Data Flow Diagram



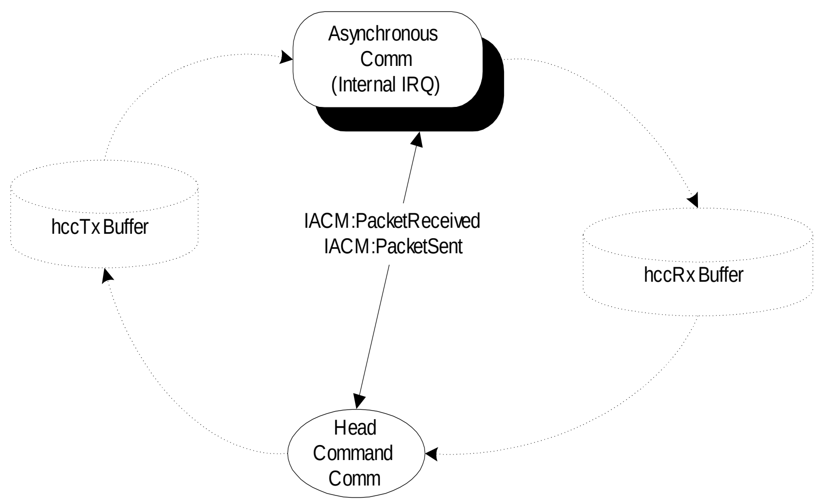
5.7. Asynchronous Comm IRQ (IACM) Task Description

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 (RS232 @ 65KBaud) Interrupts that will be generated by the incoming replies of the Head Scanner.

This ISR sends what is in the Tx Buffer and receives using the Rx buffer. HCC owns the two communication buffers. HMM uses access functions to set or retrieve these buffers.

5.7.1. IACM : HCC - Data Flow Diagram



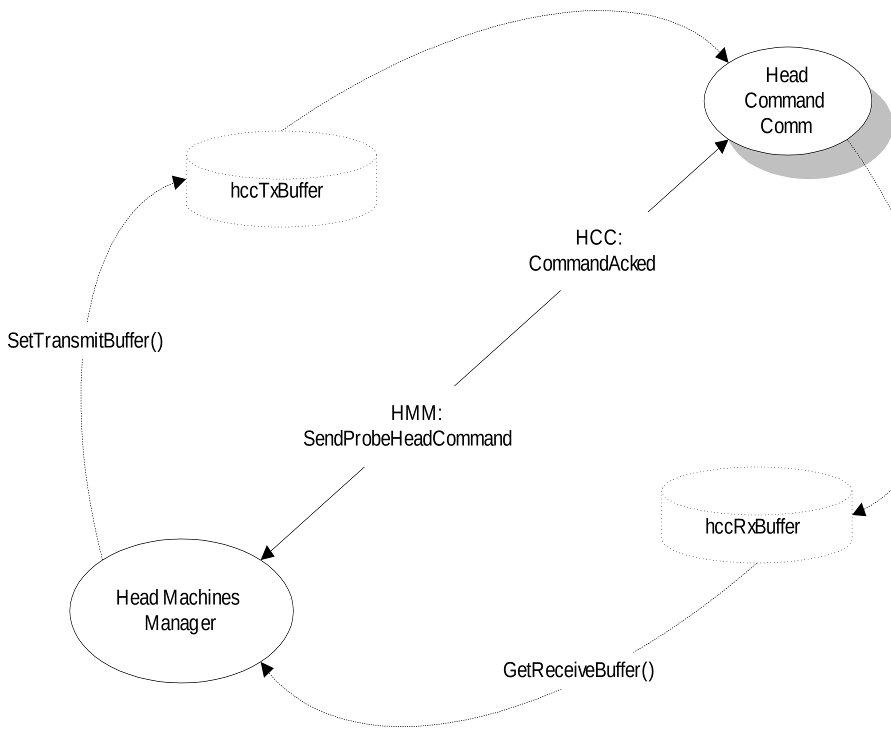
5.8. Head Command Comm (HCC) Task Description

HCC - Head Command Comm

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. The task will include Timeout detection where the timing services of the OS Kernel will be utilized.

The Head Scanner is a slave of the Color Smart PCI Card and will only respond when polled.

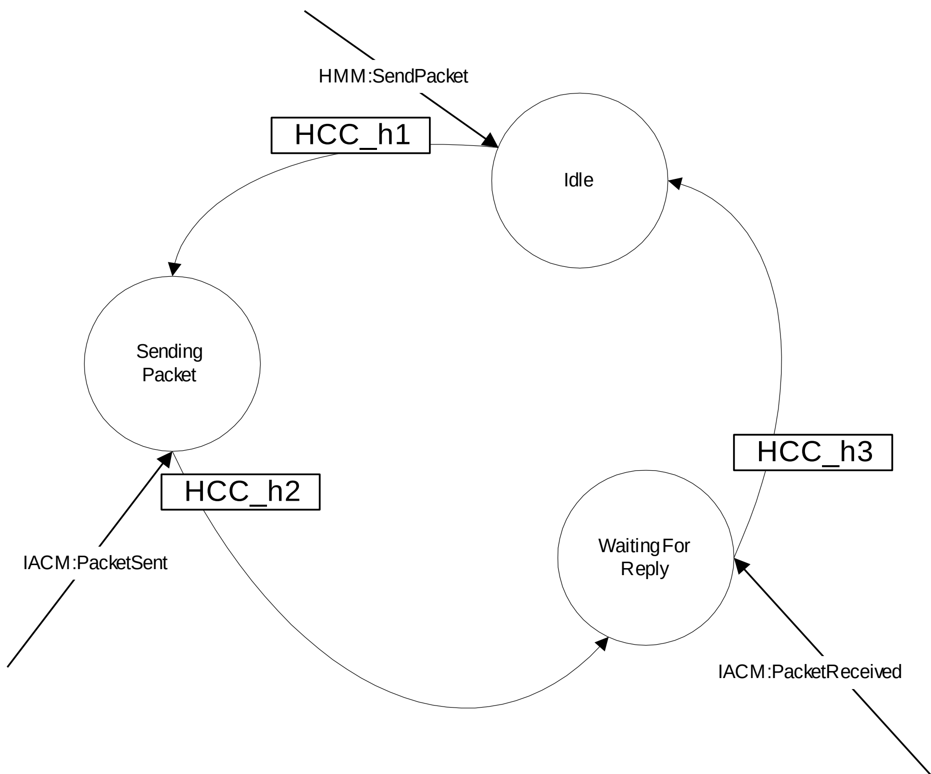
5.8.1. HCC – High Level Context Diagram



5.8.1.1. HCC – SendPacket, State Transitions

A command is sent to the head by first filling up the TXBuffer and then sending a SendPacket message to HCC. The SendPacket message is request only made by HMM.

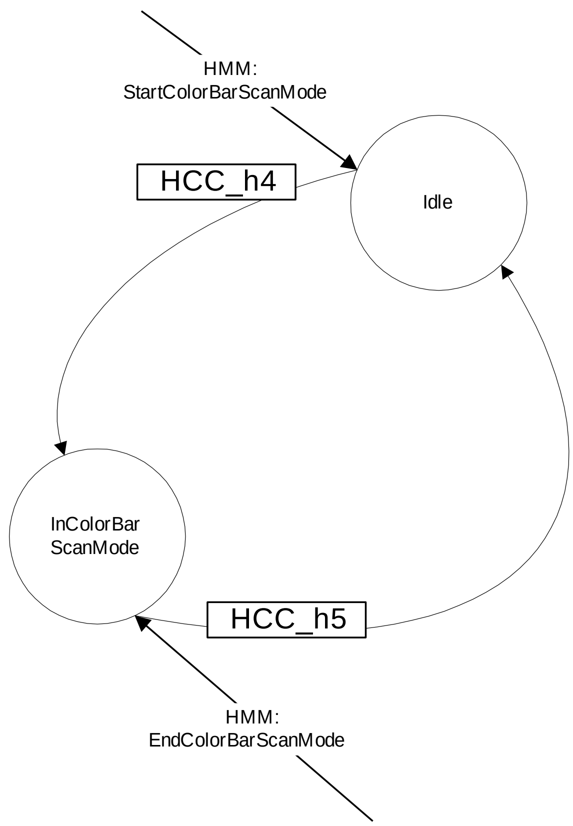
The head communication protocol specifies that each command to the head will generate a reply. This is why the state transitions from Sending directly to Waiting For a reply.



5.8.1.2. HCC – StartColorBarScanMode, EndColorBarScanMode - 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 EndColorBarScanMode message.

To limit latencies induced by the Kernel, we will be sending the "Measure Command" at the interrupt level. See Motor Chip Set Comm (IMCSC) on page 57 for more detail.



5.9. 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 121 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.

5.10. Motor Comm (MTRC) Task Description

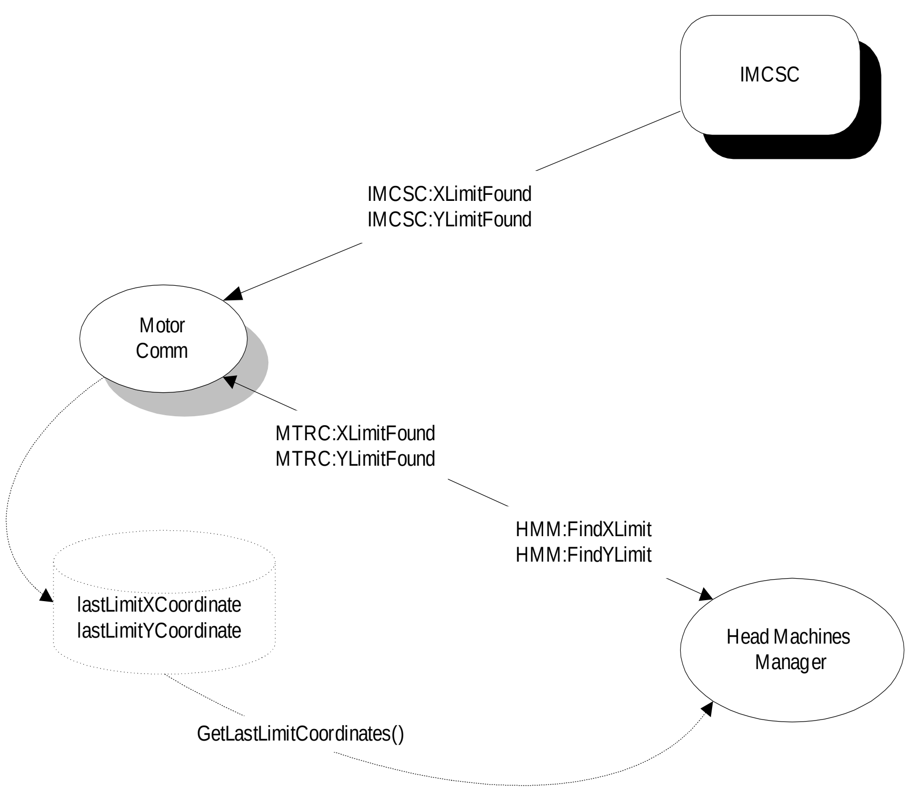
MTRC - MoToR Comm

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.

5.10.1.1. MTRC – System Interaction, FindLimits

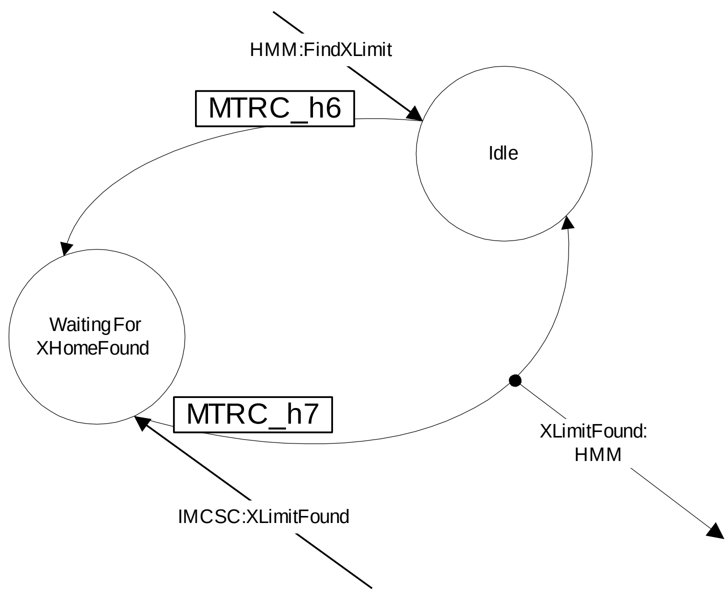


5.10.1.1.1. MTRC – FindXLimit, State Transitions

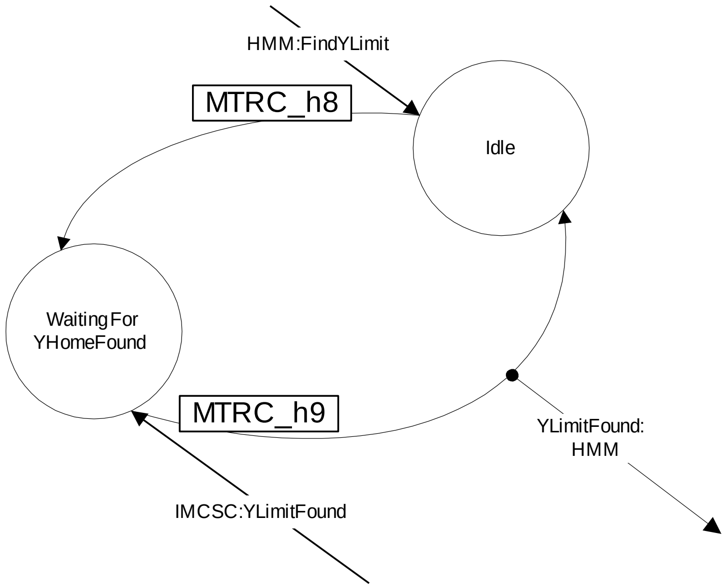
Once MTRC receives the “Find Limit” message request, it makes all the necessary motor commands to force the probe head to hit the limit and stop. The machine stays in the “Waiting” state while the probe head is moving towards the limit. A “Limit Found” message takes the machine back to idle. HMM is notified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



5.10.1.1.2. MTRC – FindYLimit, State Transitions



5.10.1.2. MTRC – System Interaction, GotoXYTarget

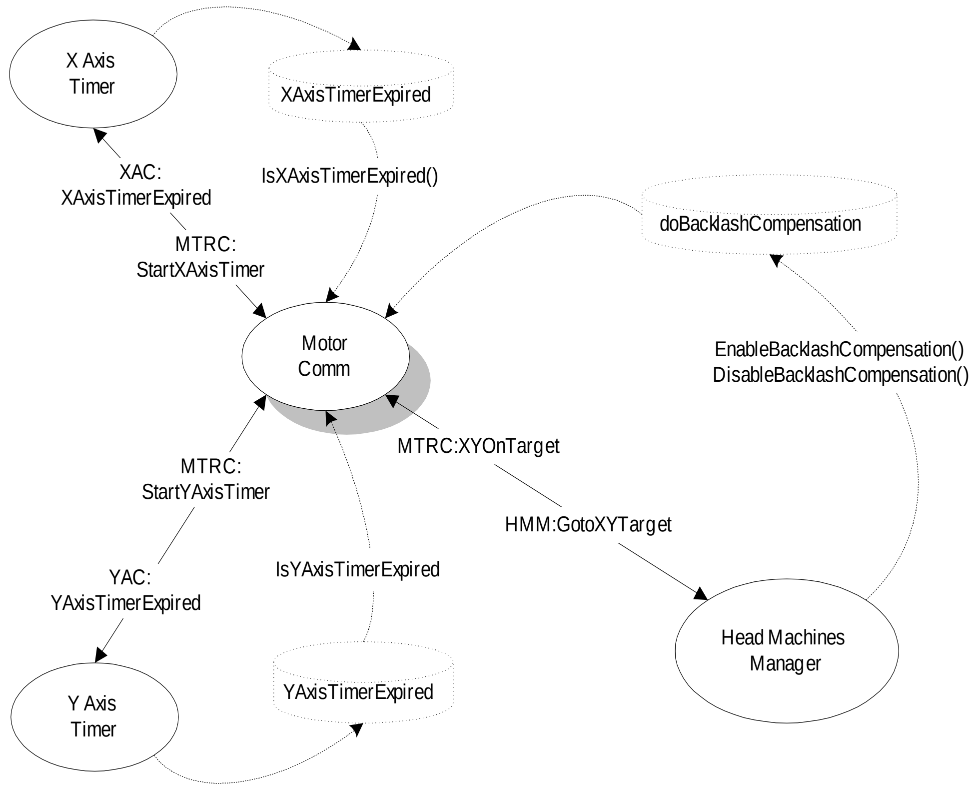
The "GotoXYTarget" is a message sent by HMM. This message is accompanied with data:

Data1 = target X Coordinate
Data2 = target Y Coordinate

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

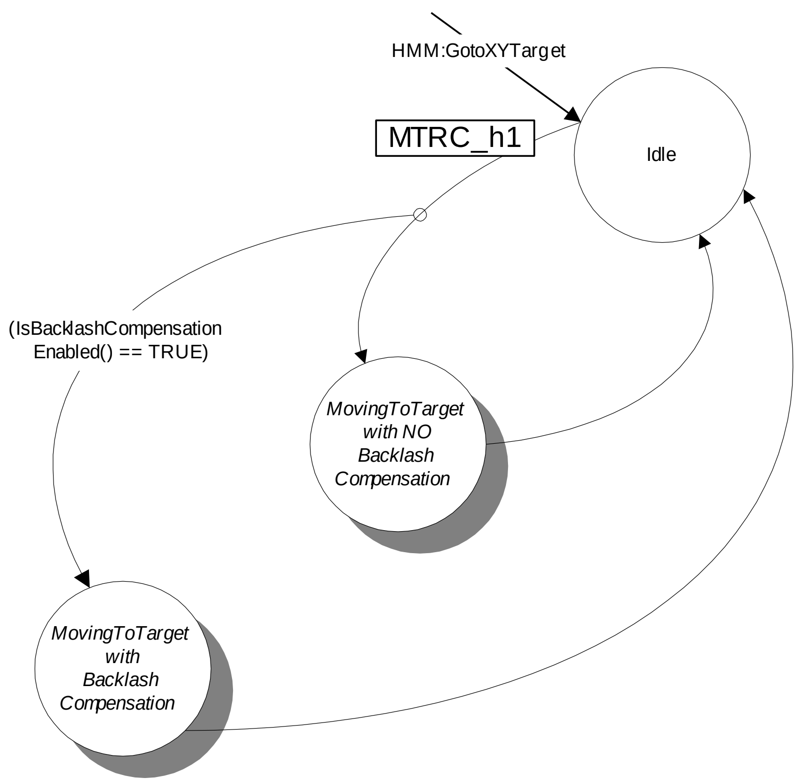
IMPORTANT:

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.


X Axis
Timer
XAxisTimerExpired
XAC:
XAxisTimerExpired
IsXAxisTimerExpired()
doBacklashCompensation
MTRC:
StartXAxisTimer
Motor
Comm
EnableBacklashCompensation()
DisableBacklashCompensation()
MTRC:XYOnTarget
MTRC:
StartYAxisTimer
HMM:GotoXYTarget
YAC:
YAxisTimerExpired
IsYAxisTimerExpired
Head Machines
Manager
Y Axis
Timer
YAxisTimerExpired

5.10.1.2.1. 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.



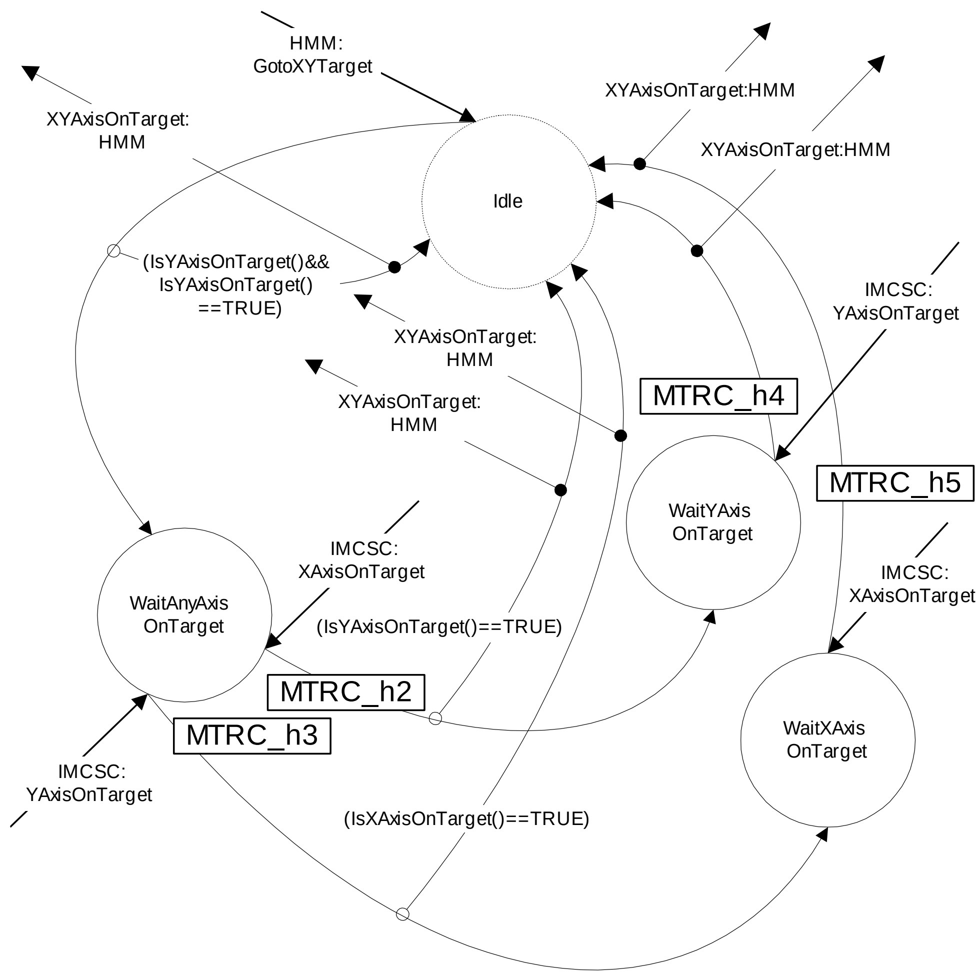
5.10.1.2.2. MTRC – GotoXYTarget, State Transitions (Backlash Compensation Disabled) – Level 2

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.

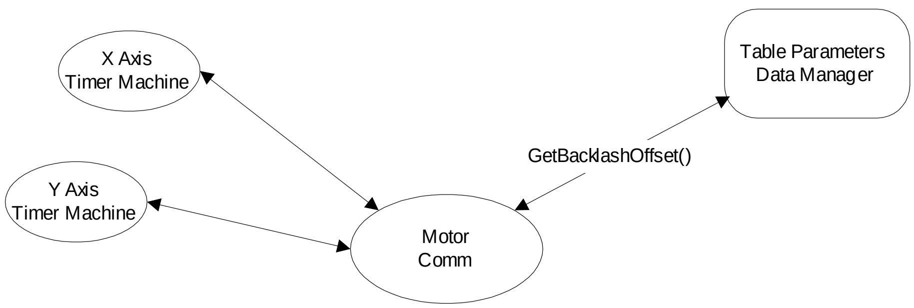
The reason for having 2 wait states, 1 for X and 1 for Y is that one axis can finish ahead of the other. The individual states accurately reflects what the SM is doing.



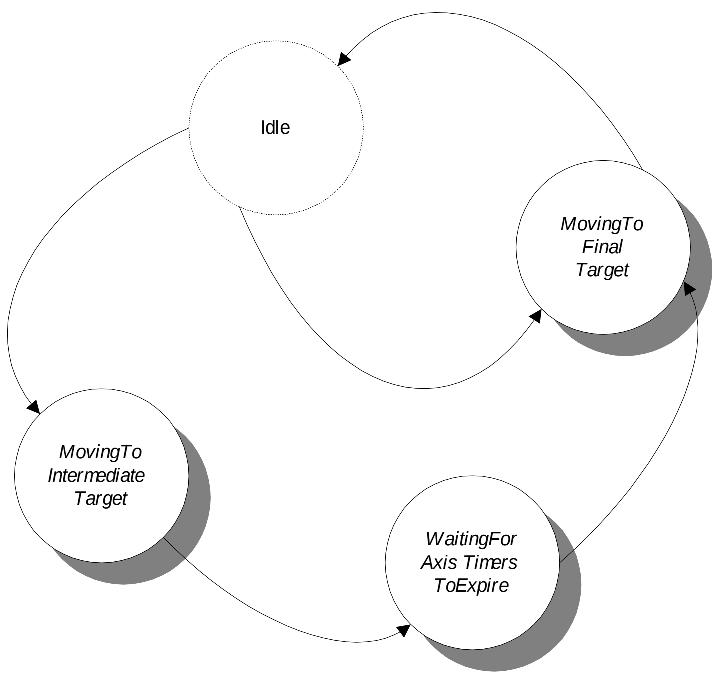
5.10.1.2.3. MTRC – GotoXYTarget, State Transitions (Backlash Compensation Enabled) – Level 2

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.



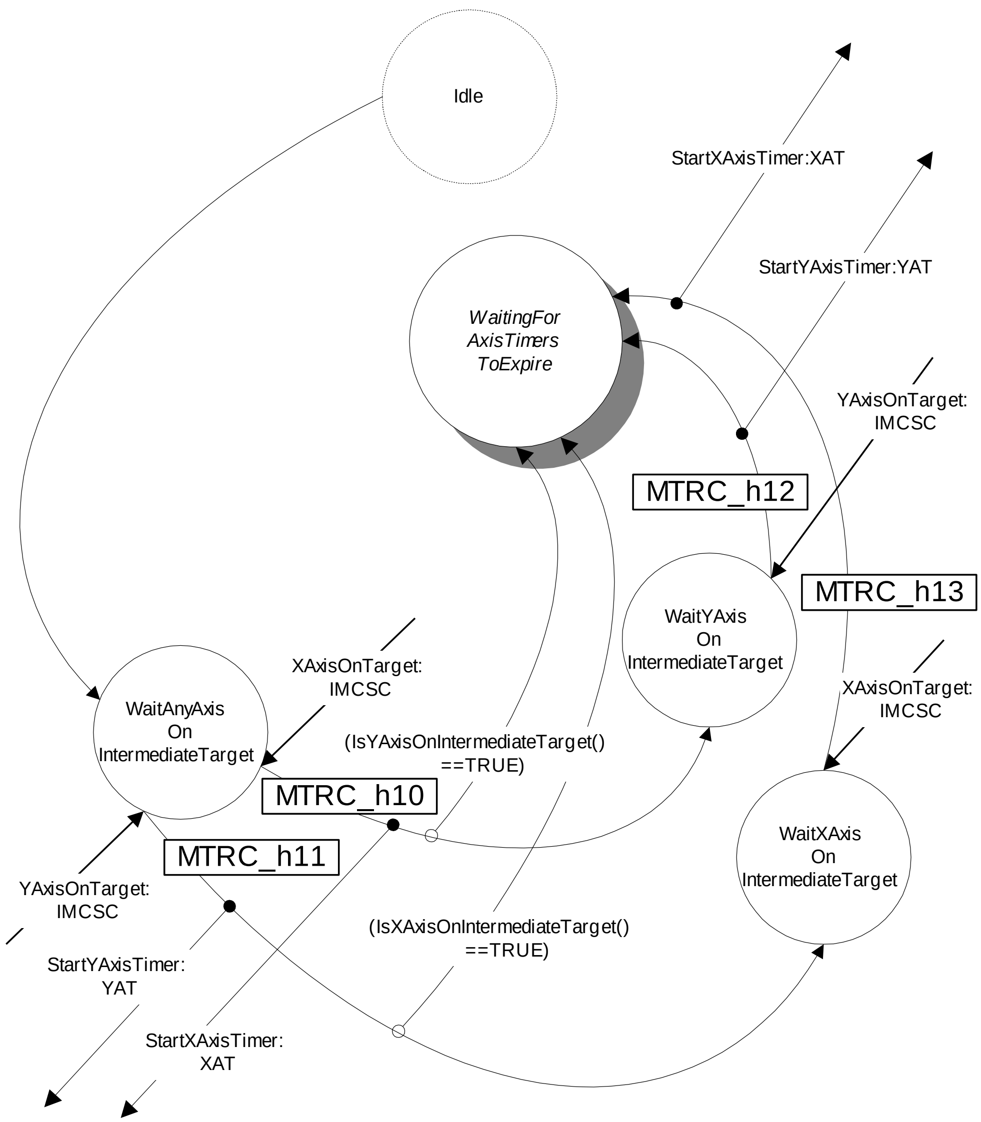
Due to the complexities of the backlash moves, the states are further divided into three groups. These groups are shown in detail in the next pages.



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

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

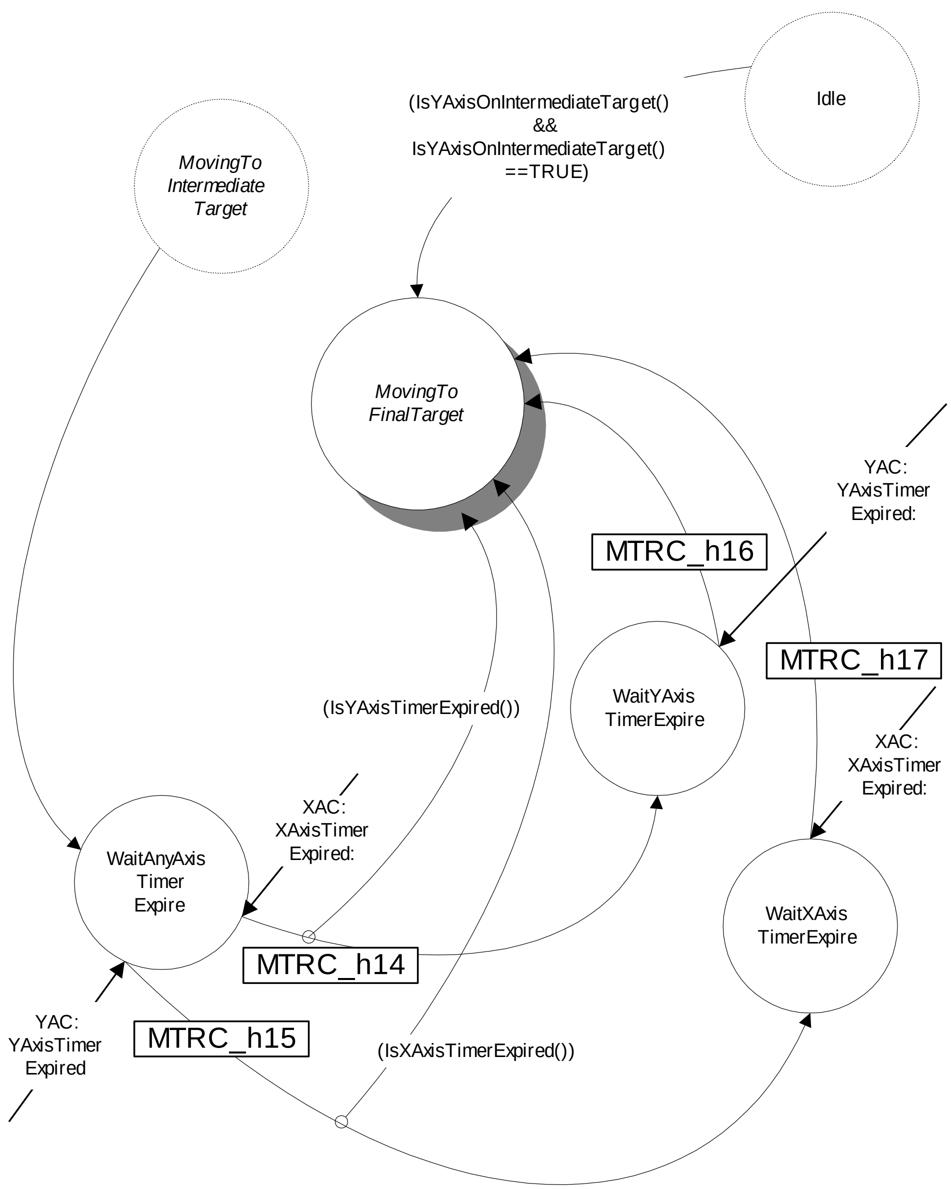
- MTRC enters the "WaitAnyAxisOnIntermediateTarget" only when a move needs to be made. Before the machine enters this state, a move command is sent for the appropriate axis.
- The backlash distance is stored by the Table Parameters Data Manager
- If both axis are already on target, the machine goes directly 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.



5.10.1.2.5. 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.

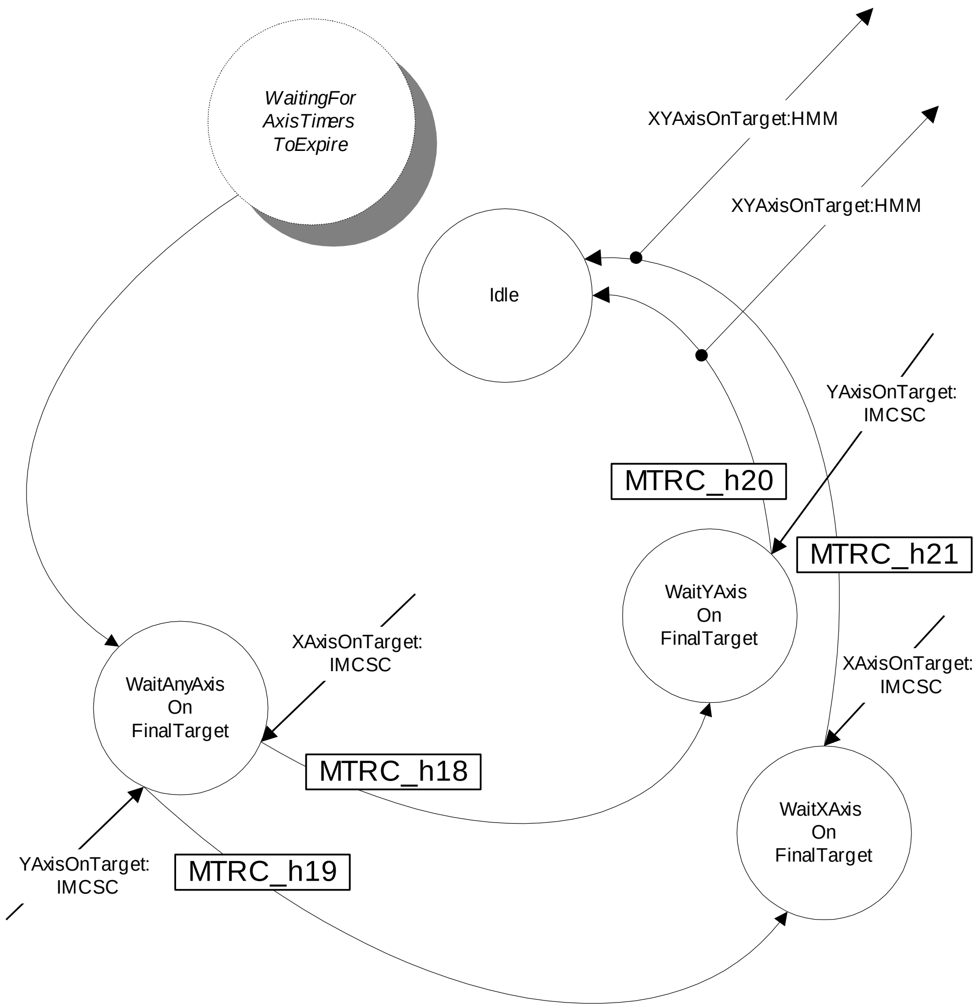
Note: If both axis are already on their intermediate target, the machine goes directly from "Idle" straight to the "Moving to Final Target" states.



5.10.1.2.6. 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



5.10.1.2.7. MTRC – GotoXTarget, GotoYTarget State Transitions

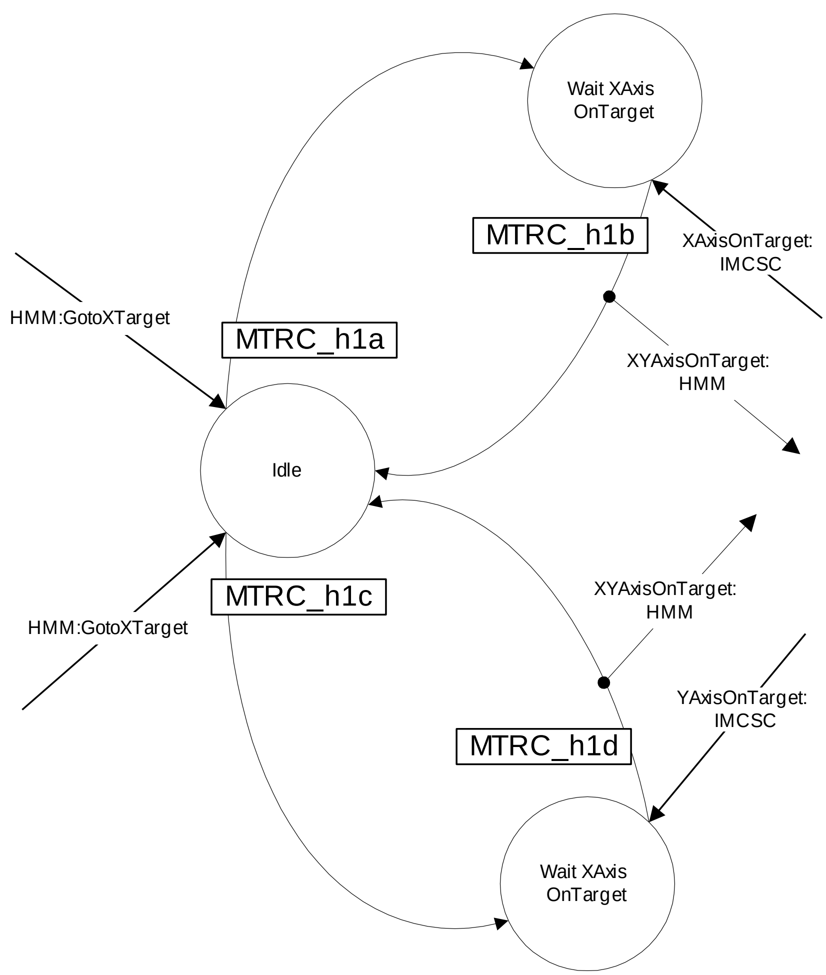
These messages allow the moving of only one axis. When a “Goto Target” command is received, MTRC sends the appropriate move command to the motor chip set to start performing the move. Once the move is complete, MTRC receives an “Axis On Target” event. It then informs HMM about the completion of the request.

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.



5.10.1.2.1. MTRC – SetNewMoveProfile, State Transitions

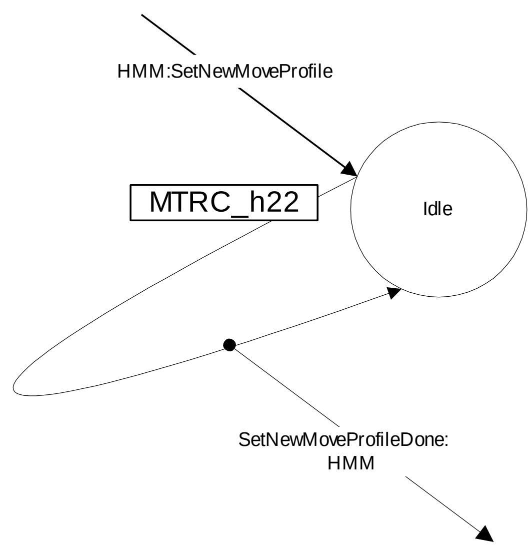
IMPORTANT:

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

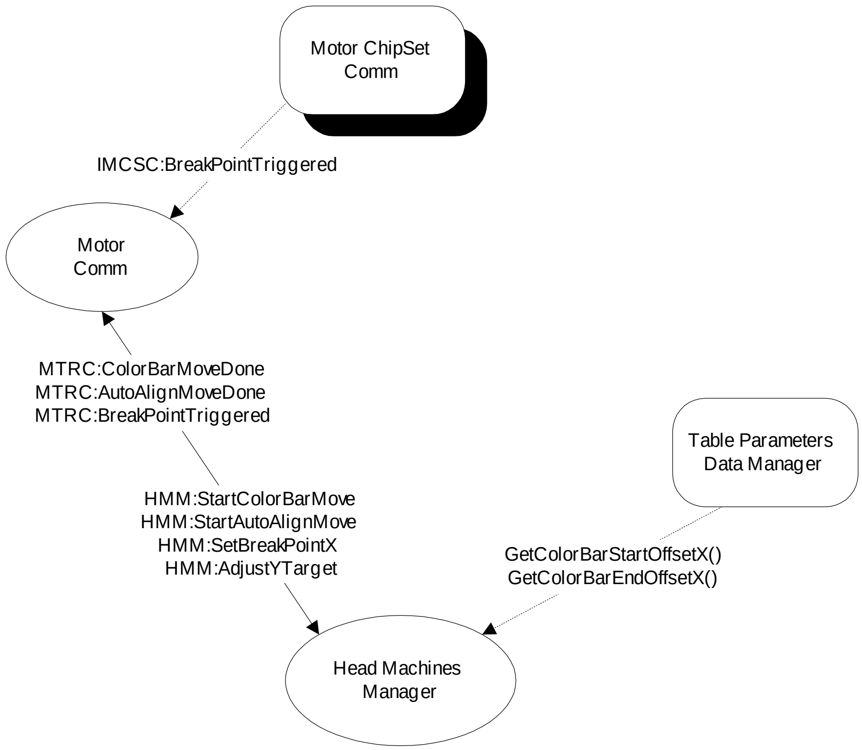
MTRC retrieves the move profile data from the Table Parameters Data Manager using the given index.



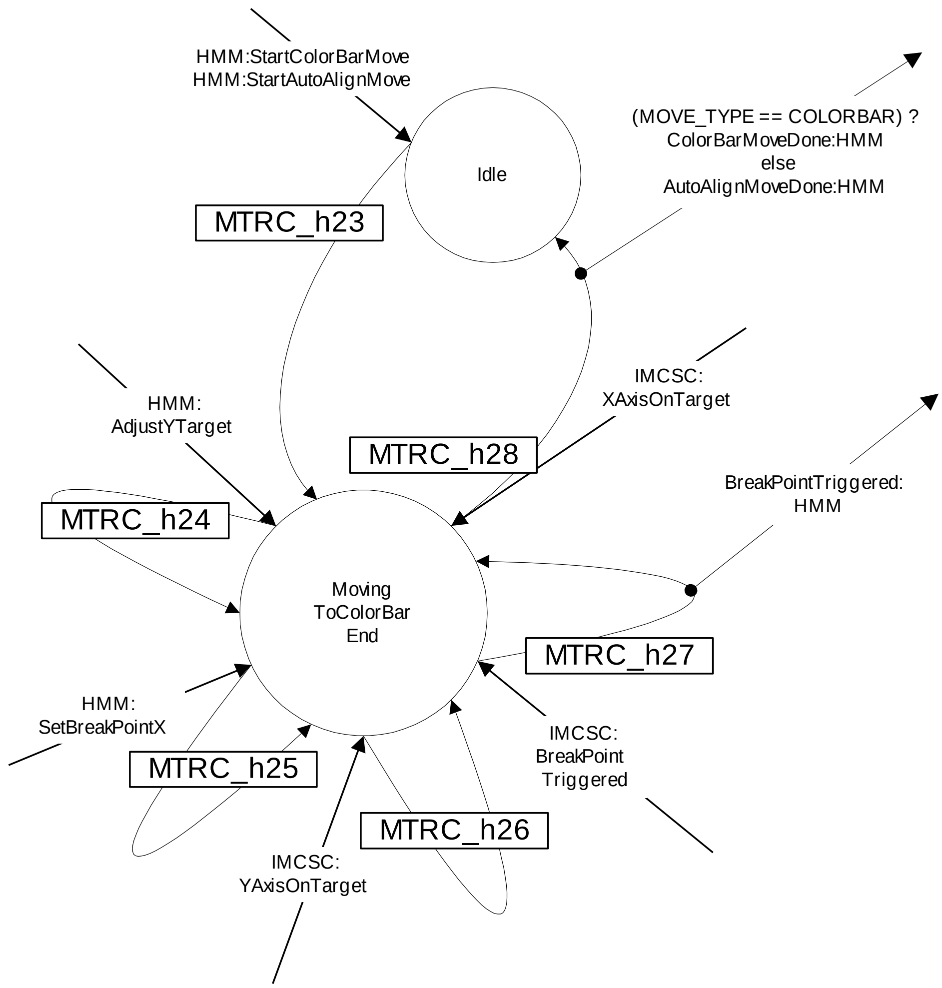
Note:

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

5.10.1.3. MTRC – System Interaction, StartColorBarMove and Start Auto Align Move



5.10.1.3.1. MTRC – Start Color Bar Move, Start Auto Align Move, State Transitions



5.10.1.4. 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.

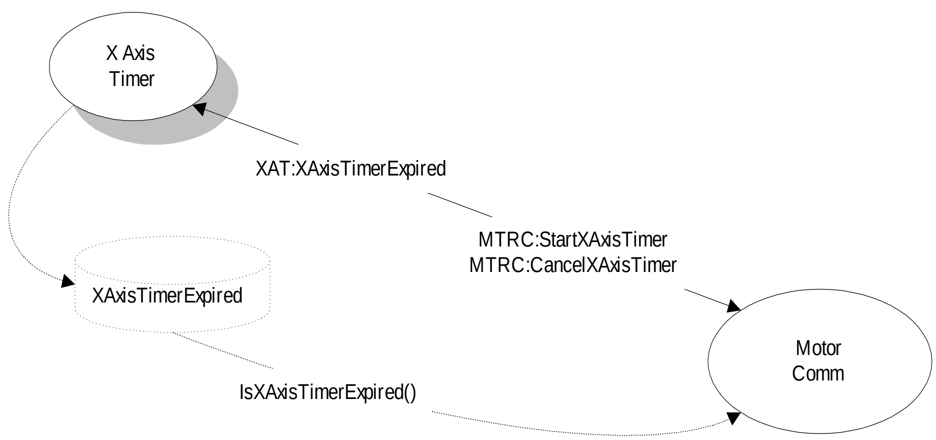
The “StartXAxisTimer” is a message request sent by MTRC. This message is accompanied with data:

Data1 = Timeout Duration in milliseconds
Data2 = NULL

IMPORTANT:

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

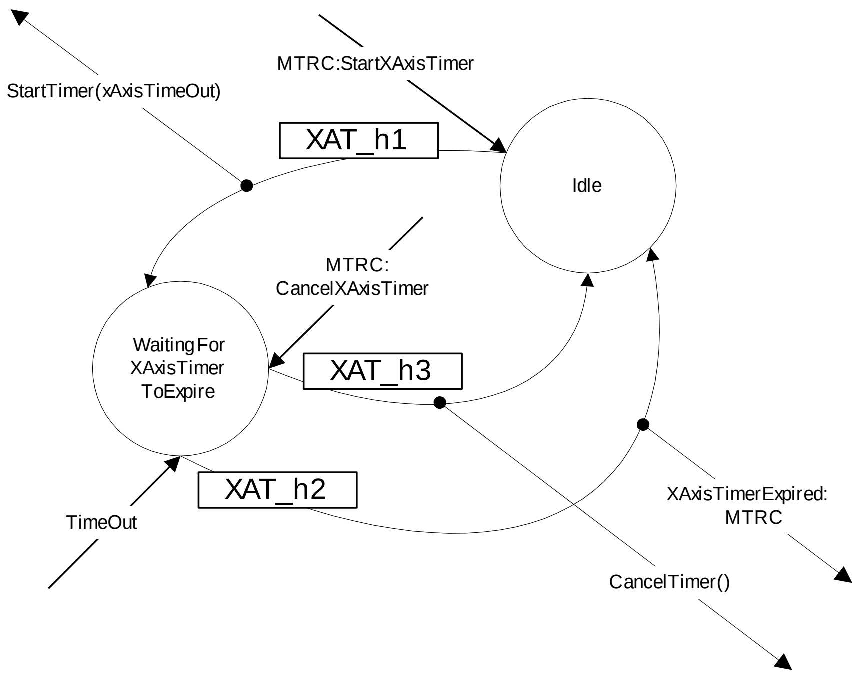
5.10.1.5. XAC – System Interaction, StartXAxisTimer, CancelXAxisTimer



5.10.1.5.1. 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.



5.10.1.6. Y Axis Timer (YAT) Task Description

***YAT – Y A**xis **T**imer*

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

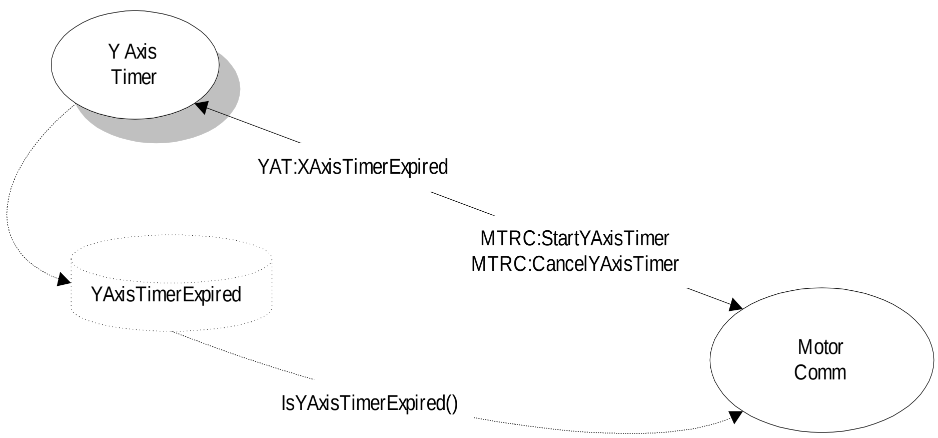
The “StartYAxisTimer” is a message sent by MTRC. This message is accompanied with data:

Data1 = Timeout Duration in milliseconds
Data2 = NULL

IMPORTANT:

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

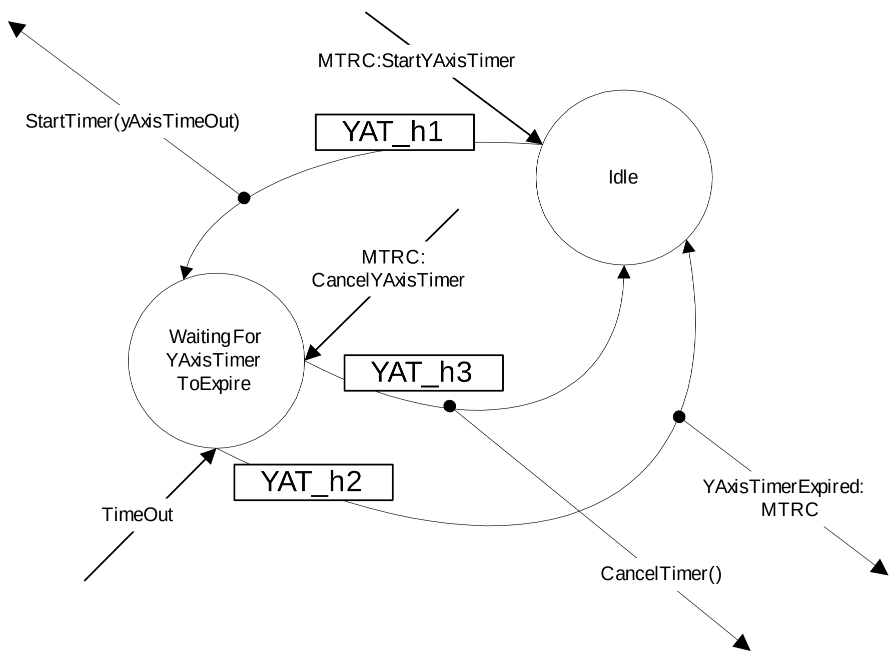
5.10.1.7. YAT – System Interaction, StartYAxisTimer, CancelYAxisTimer



5.10.1.7.1. 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.



5.10.1.8. Target Lamp Manager (TLM) Task Description

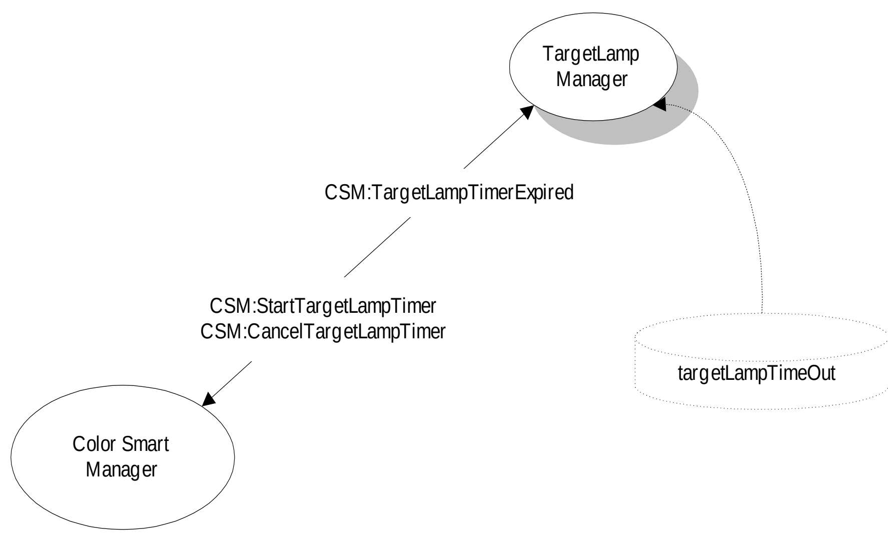
TLM – T*arget* ***L****amp* ***M****anager*

The target lamp needs to be turned off after some given time of inactivity. This SM keeps track of the amount of time that the Target Lamp has been turned on. CSM uses TLM to make sure that the lamp is lighted only for a given duration when the Probe Head is inactive.

Note: The only time that TLM will be need is when CSM is in either one of the 2 modes:

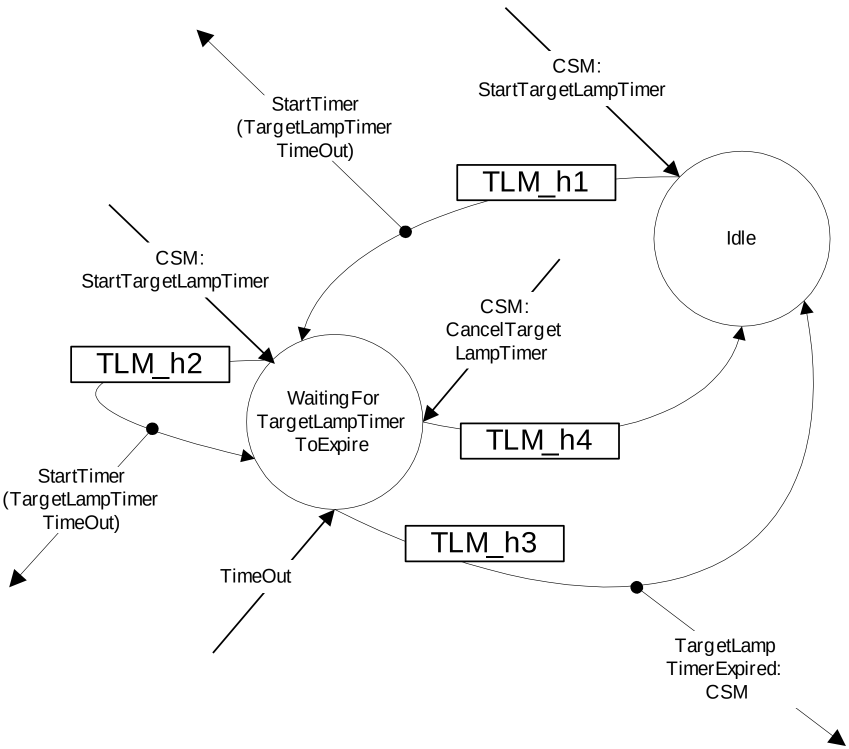
1. Track Ball Mode
2. Service Diagnostic Mode

5.10.2. TLM – High Level Diagram



5.10.2.1.1. TLM – StartTargetLampTimer, CancelTargetLampTimer, State Transitions

D1 contains the Target Lamp Timeout



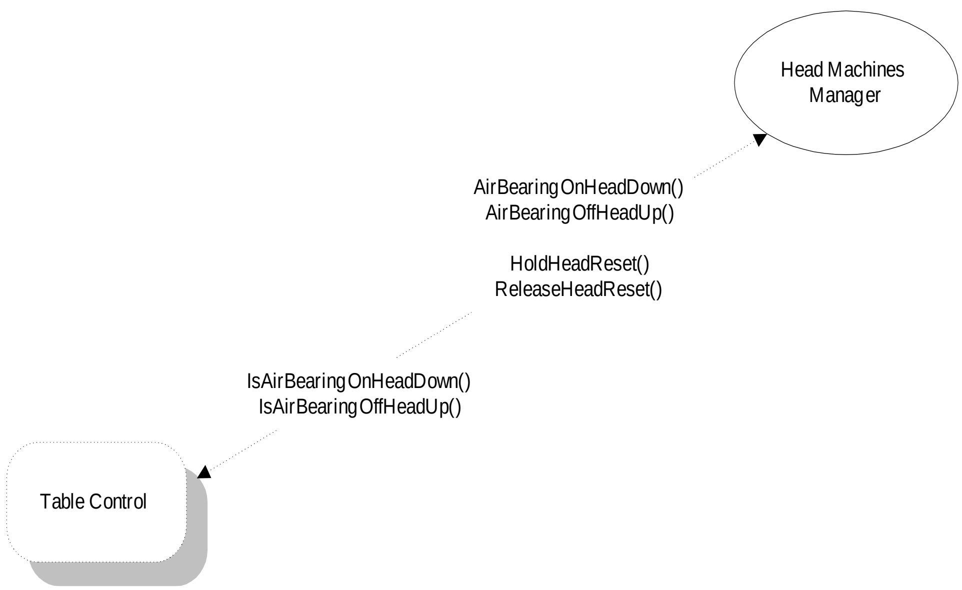
5.11. Table Control (TC) Task Description

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

The table attribute for controlling air did not seem to belong to anybody so diagram specifies that it so. In general these are just a set of functions or interface to allow the table air to be turned on or off. In addition a separate RESET line has been provided to force the head to reset itself. TC also provides this service.

Since the Head coordinator manages TC, its operation will be synchronized with the rest of the system.

5.11.1. TC – High Level Context Diagram



5.12. Head Operations Coordinator (HOC) Task Description

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

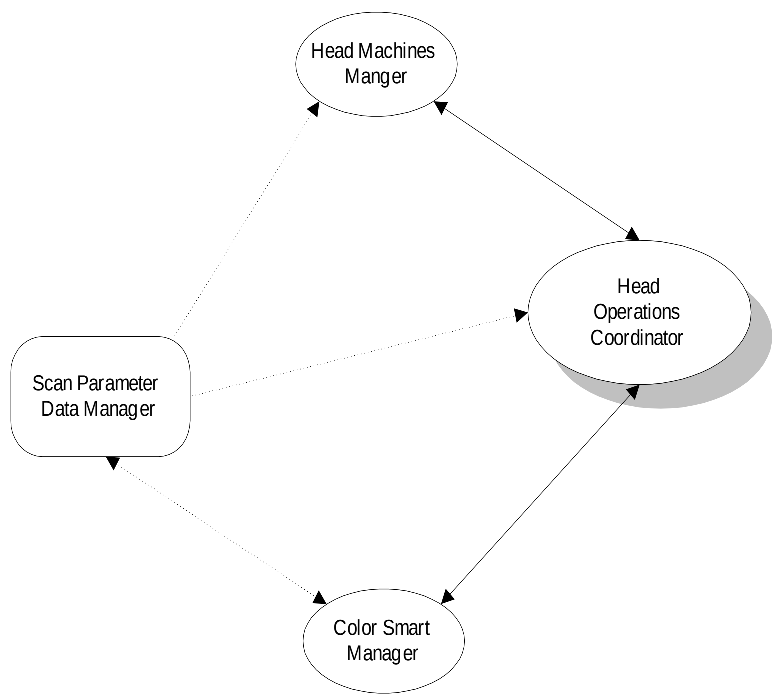
This SM was created to simplify the state transitions of both HMM and CSM.

- 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)
- 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.

5.12.1. HOC - High Level Context Diagram

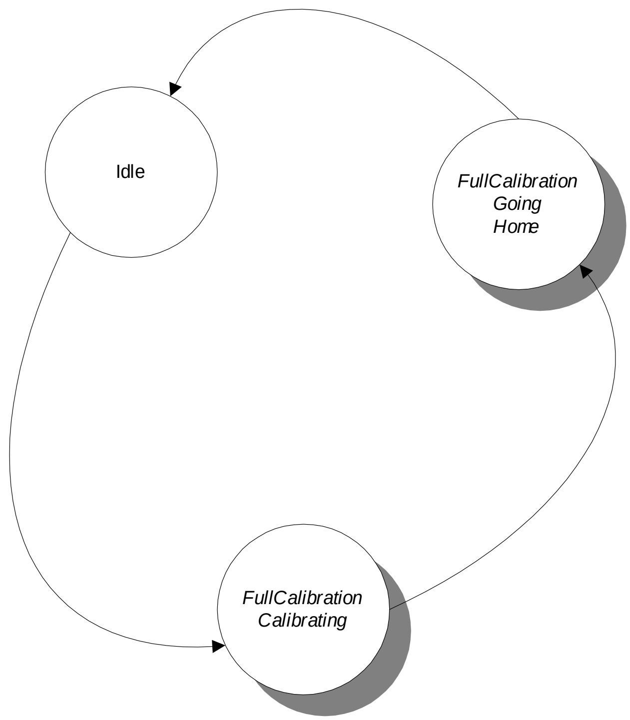
As seen below, HOC interacts with many other machines within the system. The diagram below shows the machines as well as the messages that are passed between them.



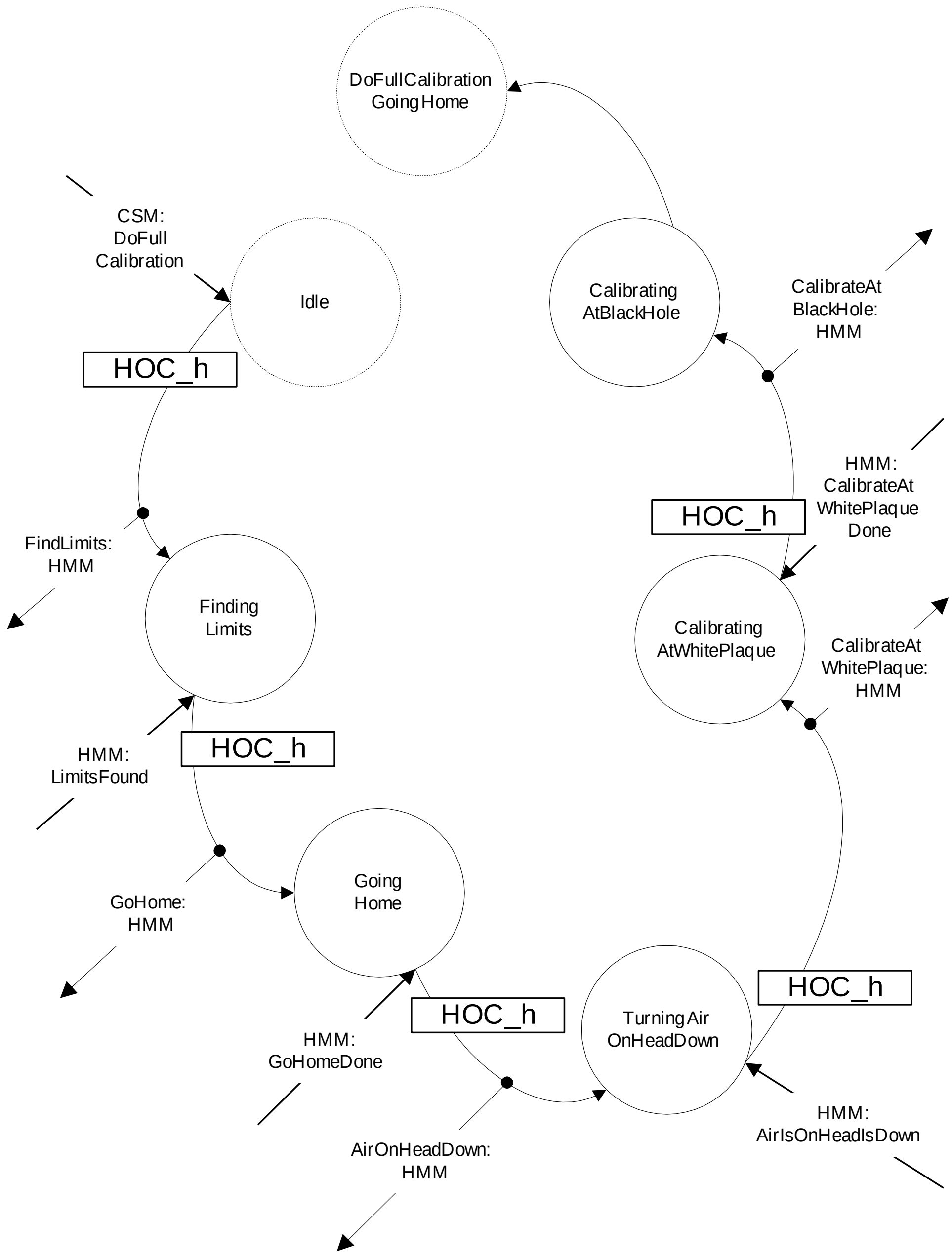
5.12.1.1.1. HOC – Do Full Calibration, State Transitions Level 1

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

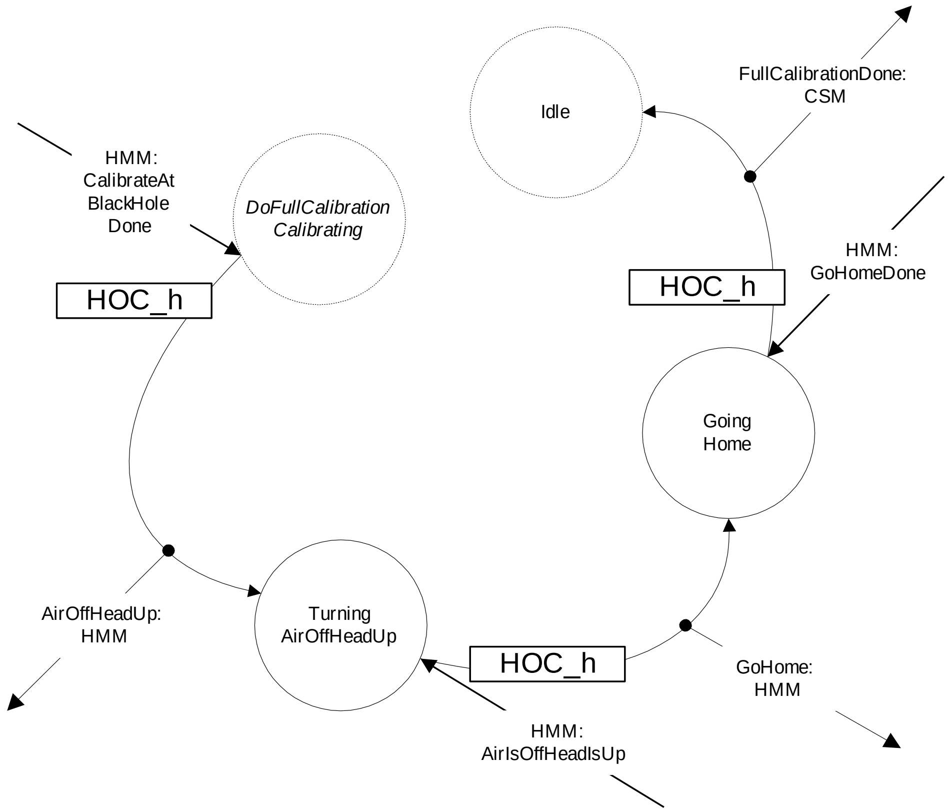
There are many sequences involved when performing this operation. This is why the different states are further shown at another level.



5.12.1.1.2. HOC –*Full Calibration Calibrating*, State Transitions Level 2

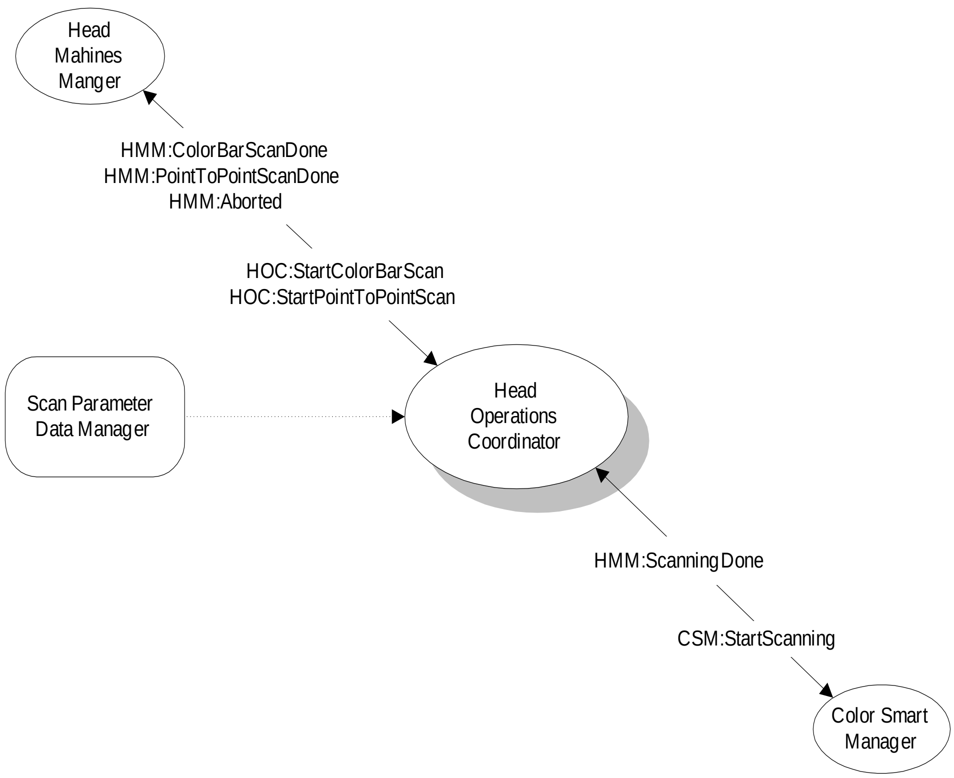


5.12.1.1.3. HOC –*Full Calibration Going Home*, State Transitions Level 2



5.12.1.2. HOC – System Interaction, StartScanning

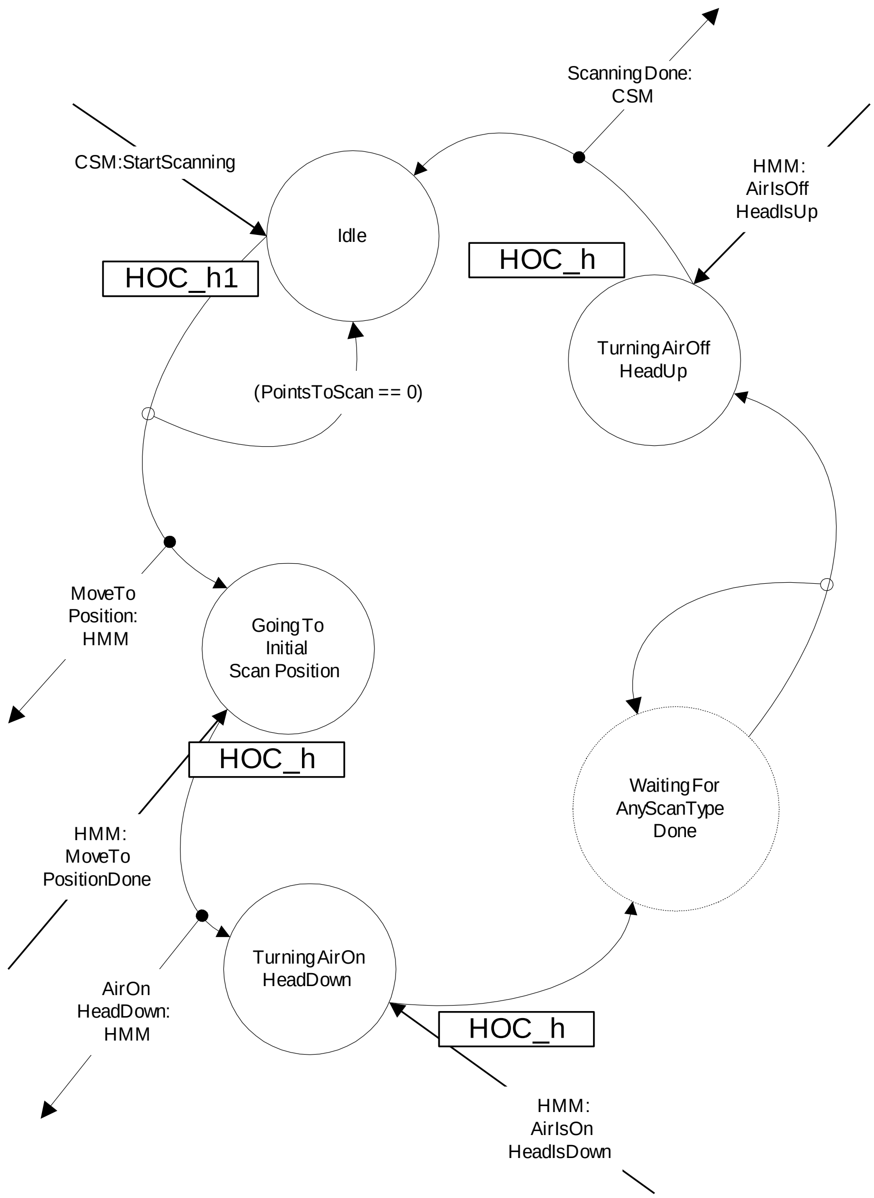
D1 contains the scanStartIndex
D2 contains the scanEndIndex



5.12.1.2.1. HOC – Start Scanning, State Transitions Level 1

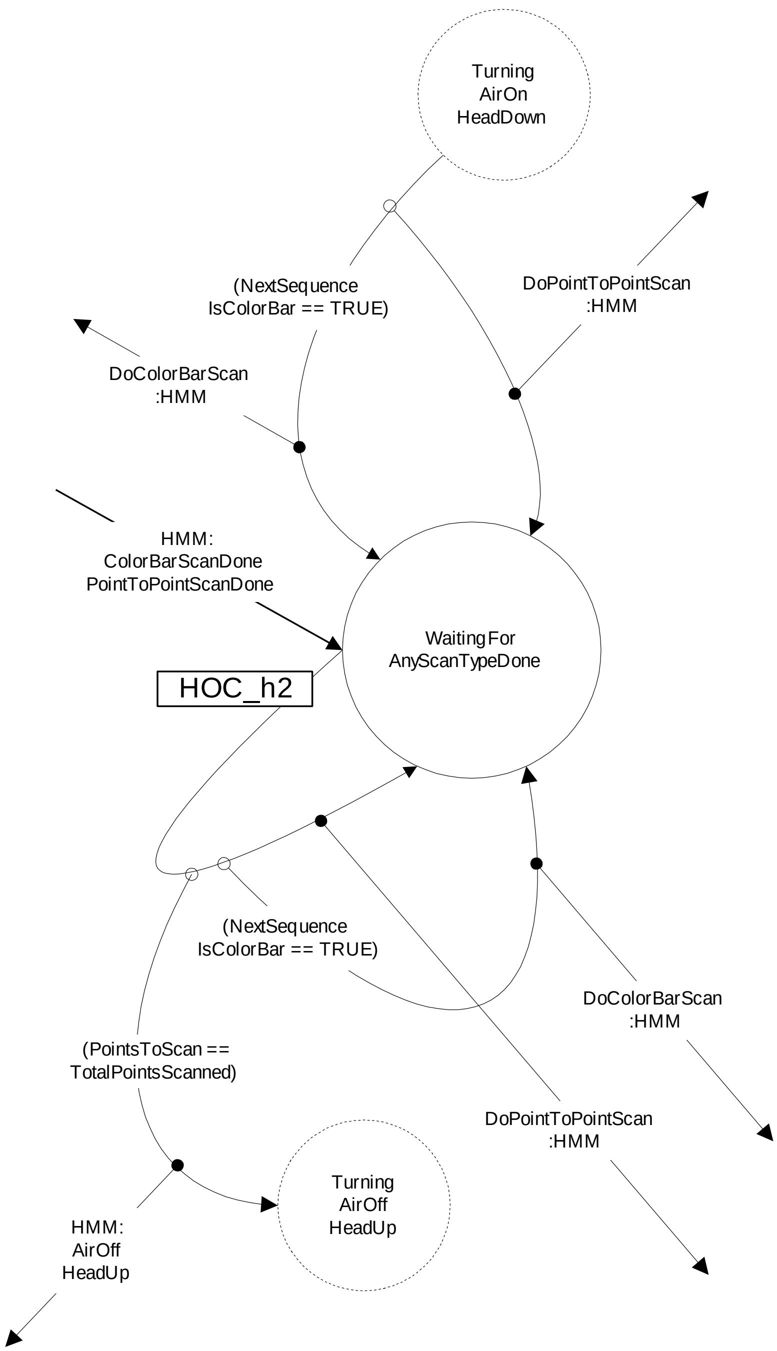
The “Start Scanning” message is generated directly by CSM.

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



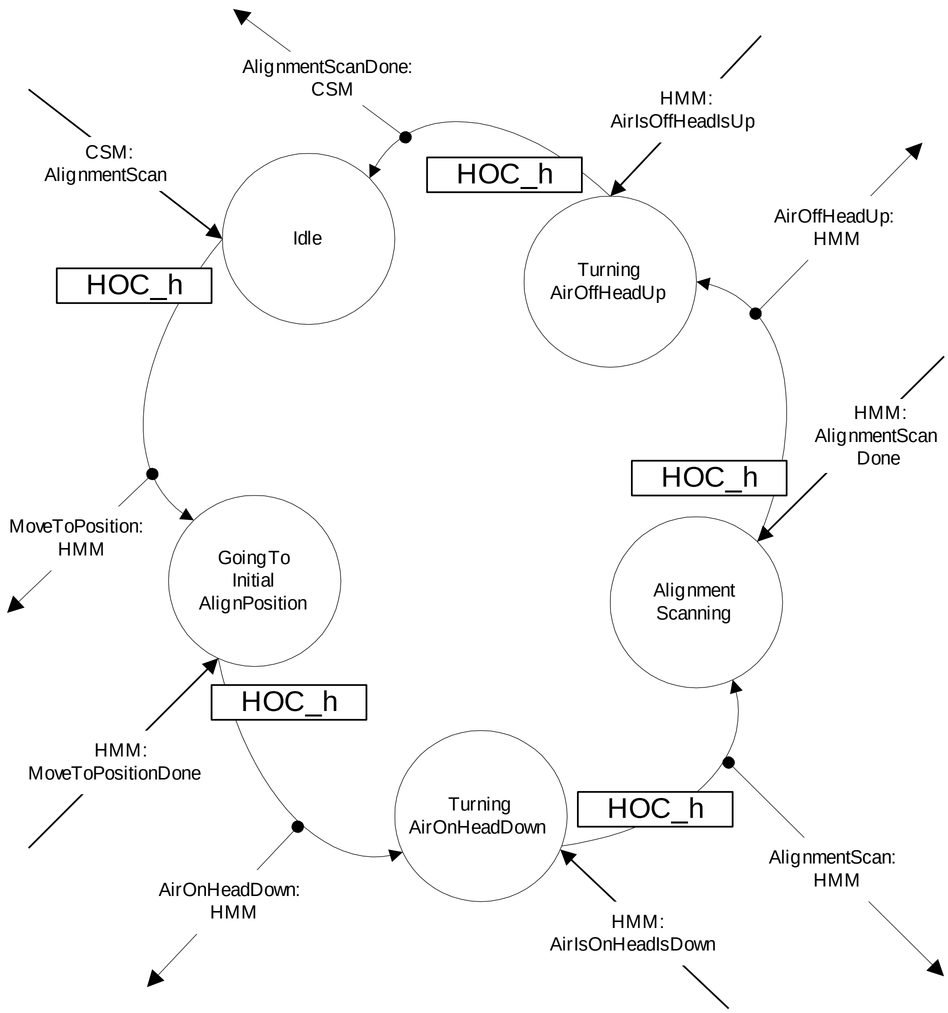
5.12.1.2.2. HOC – Waiting For Scan Selected, State Transitions Level 2

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



5.12.1.2.3. HOC – AlignmentScan, State Transitions

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



5.13. Head Machines Manager (HMM) Task Description

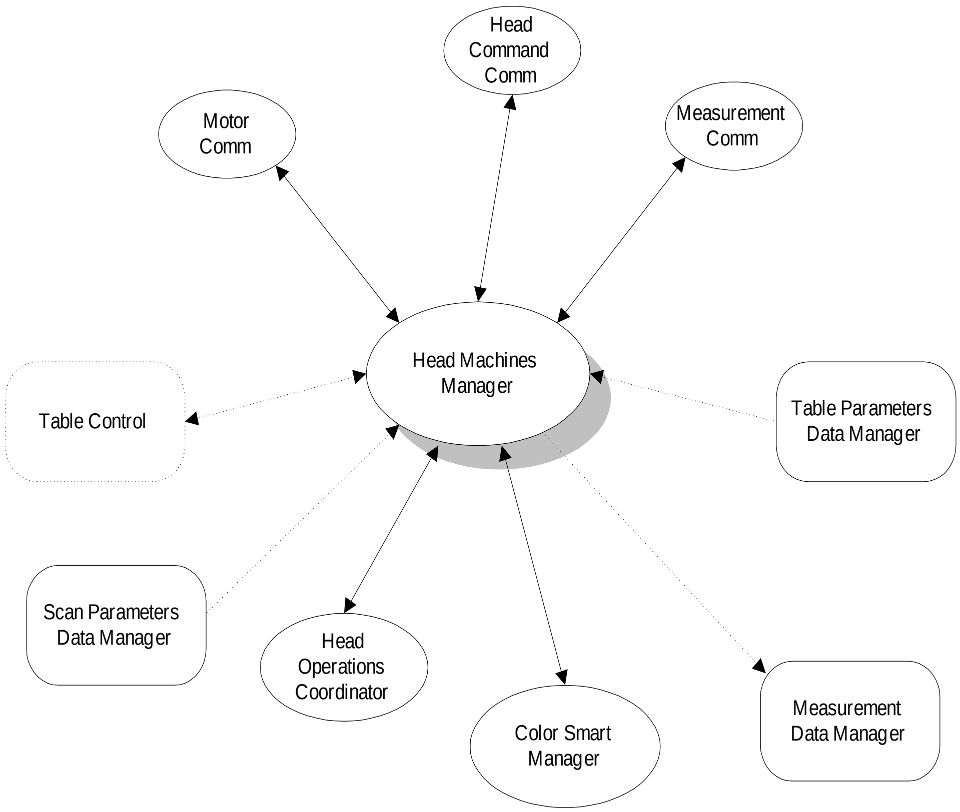
***HMM - H**ead **M**achines **M**anager*

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:

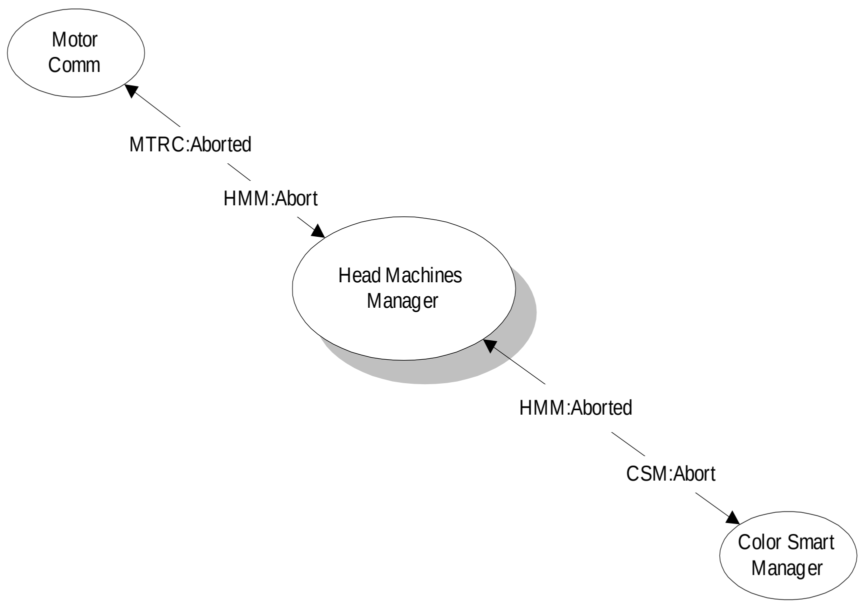
- He knows the general state of the head, including its position in the table
- We will allow short commands from Color Smart NT App such as “Turn On/Off Light”. Such a command will be execute if appropriate.
- He will only respond to a Color Smart NT App command if the command is appropriate for the 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

5.13.1. 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.

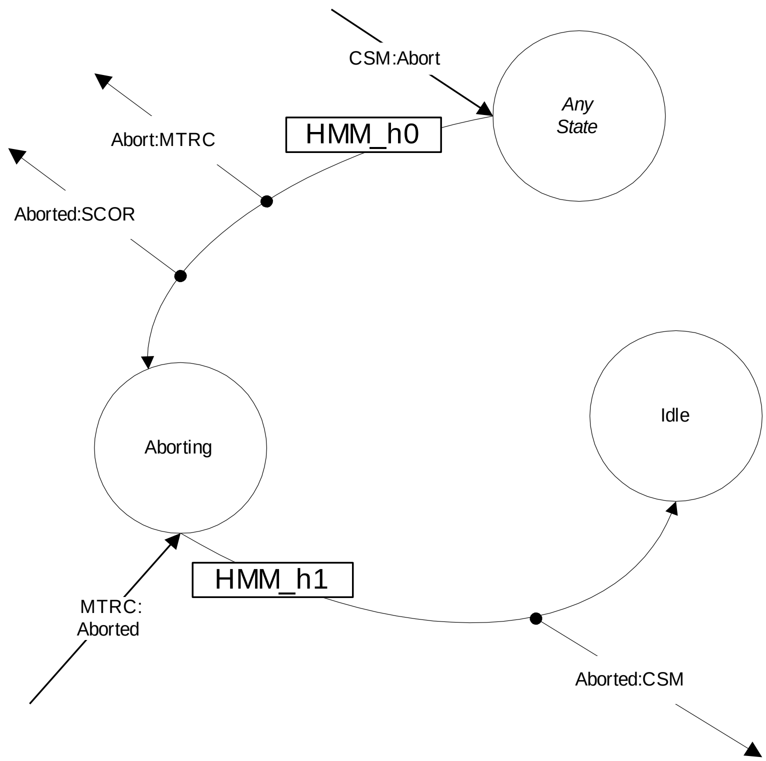


5.13.1.1. HMM – System Interaction, Abort

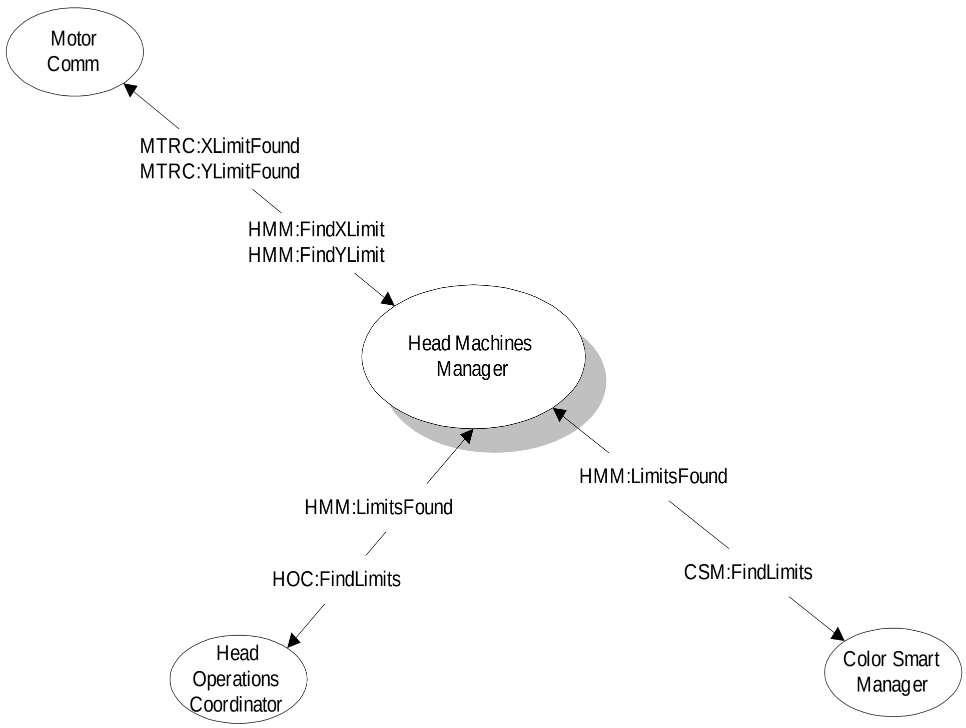


5.13.1.1.1. HMM – Abort, State Transitions

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



5.13.1.2. HMM – System Interaction, Find Limits



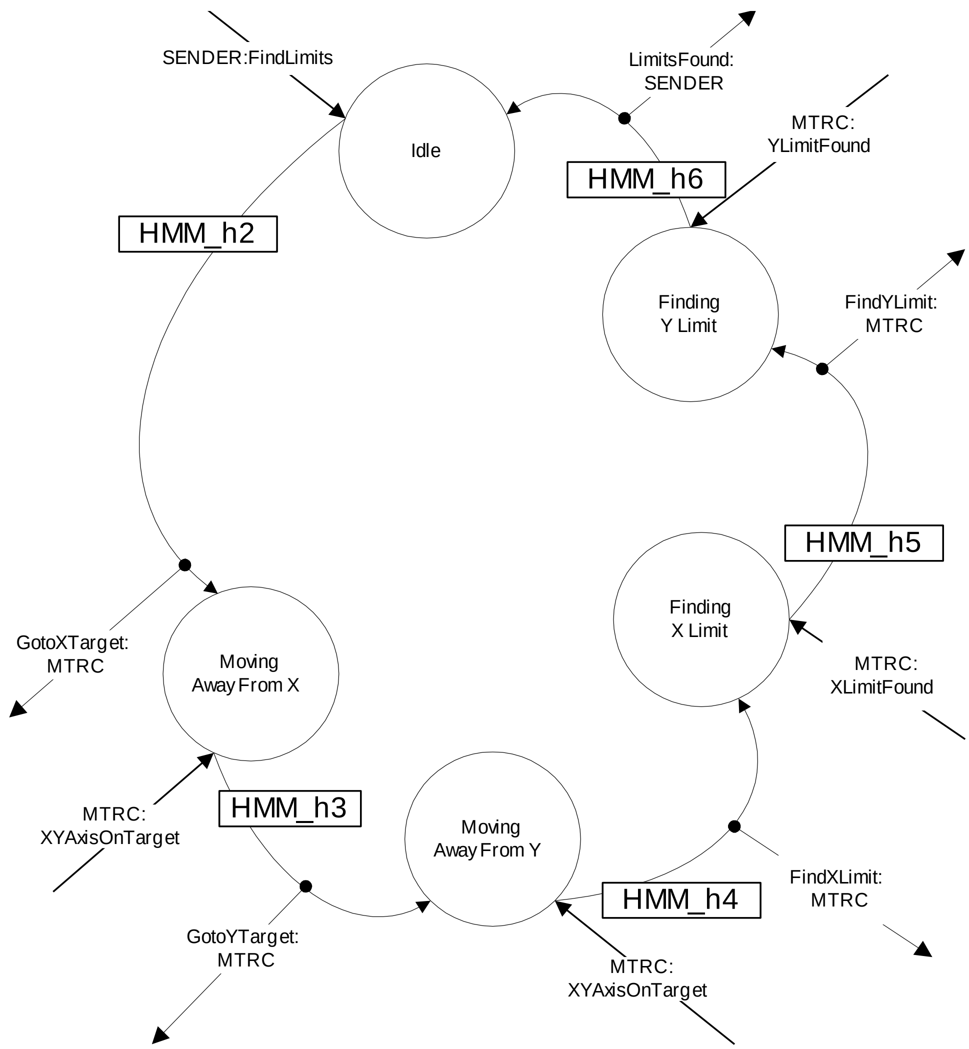
5.13.1.2.1. HMM – Find Limit, State Transitions

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.

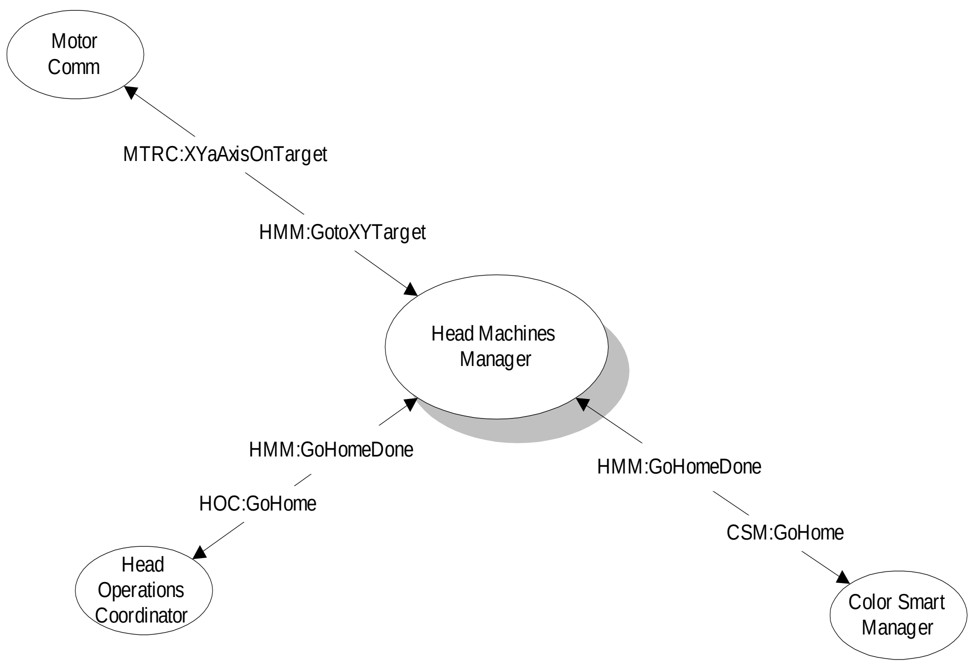
Once these steps are complete a request is sent to MTRC to “FindXLimit”. He waits until MTRC sends back a “XLimitFound” message. We repeat the same process for the Y-axis. Once the Y Limit has been also been found, reply to the SENDER telling him “LimitsFound”. HMM then returns back to the Idle State.

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



5.13.1.3. HMM System Interaction, Go Home

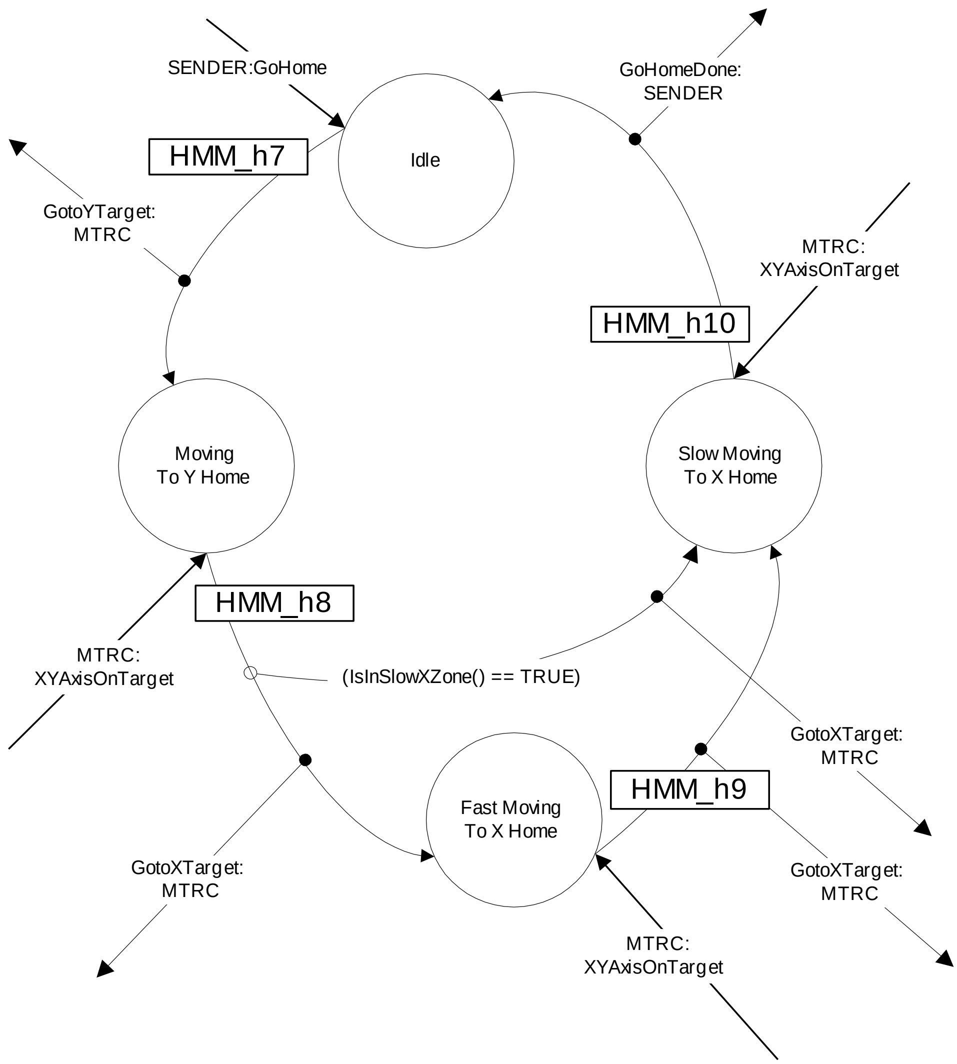
At the end of a scan or the “Trackball” modes, the head is requested to go home and be parked at the white plaque. CSM or HOC directly sends the “Go Home” message. The center of the white plaque is considered to be “Home”. HOC uses this message when it needs to perform a Full Calibration procedure.



5.13.1.3.1. 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.

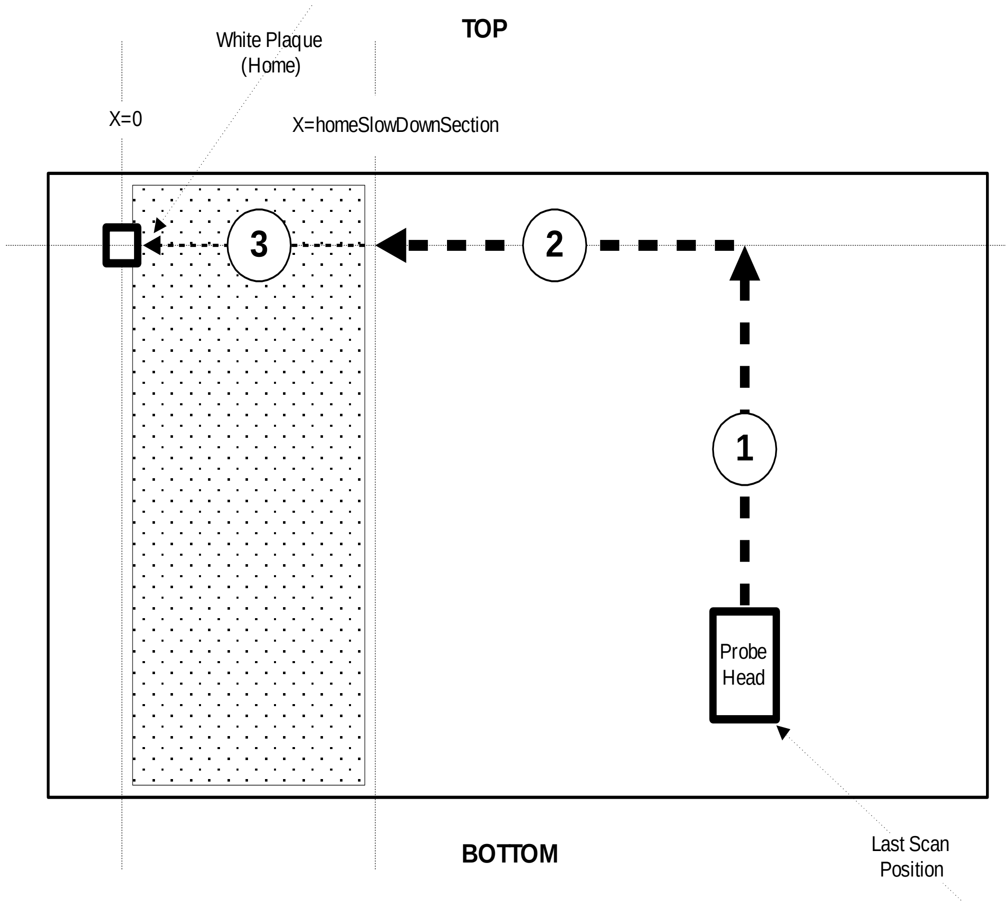


NOTE:

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

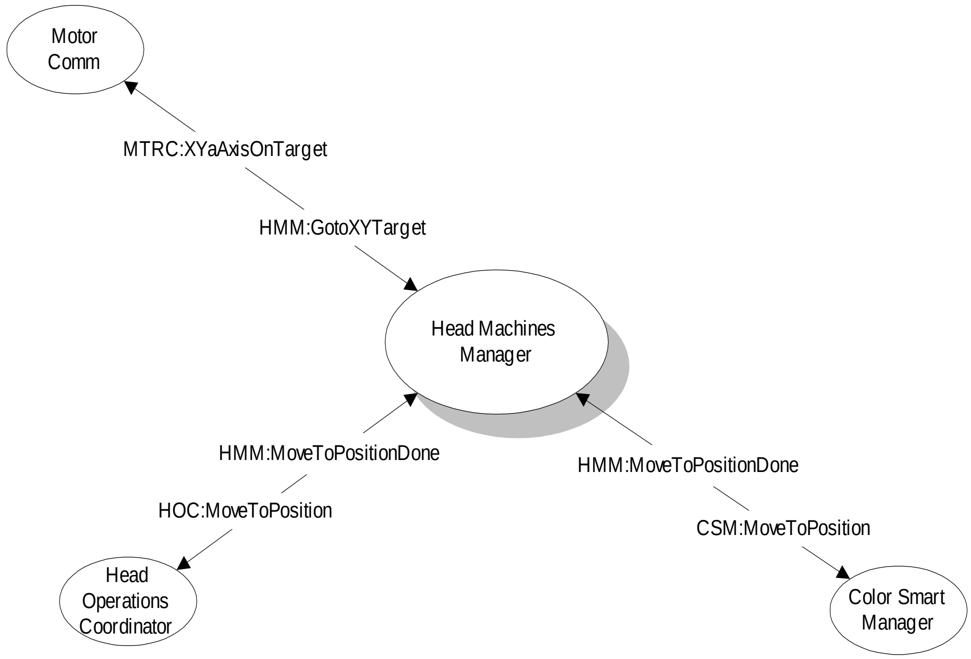
5.13.1.3.2. HMM - Go Home, Mechanical Behavior

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:



5.13.1.4. HMM System Interaction, Move To Position

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



5.13.1.4.1. HMM – MoveToPosition, State Transitions

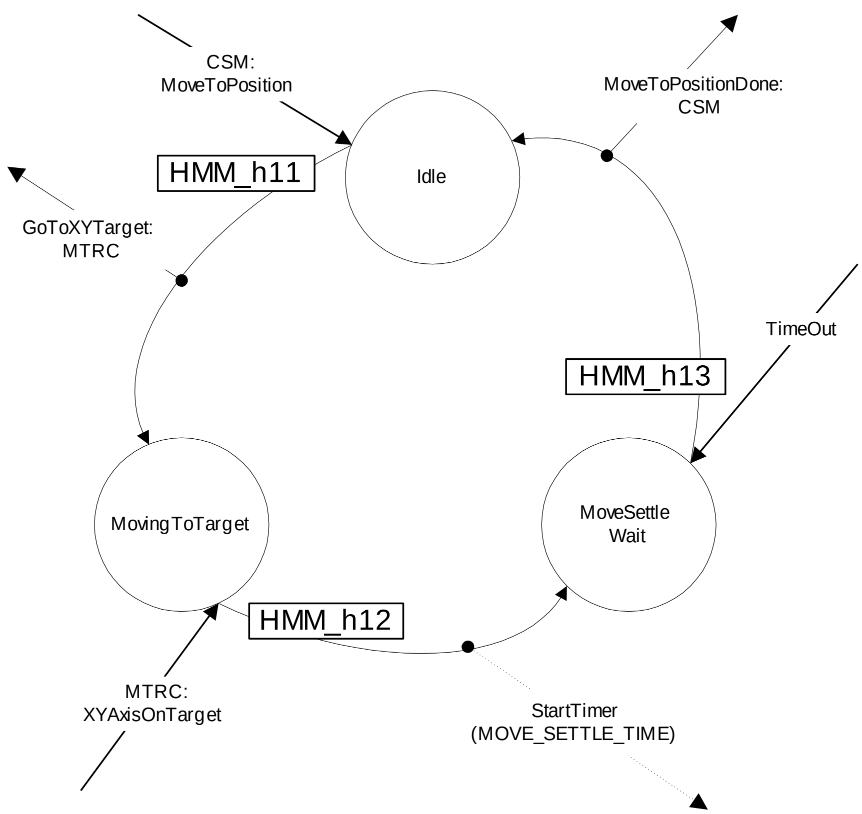
The “Move To Position” message is accompanied by data.

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 a “Measure On The Spot” command if needed.

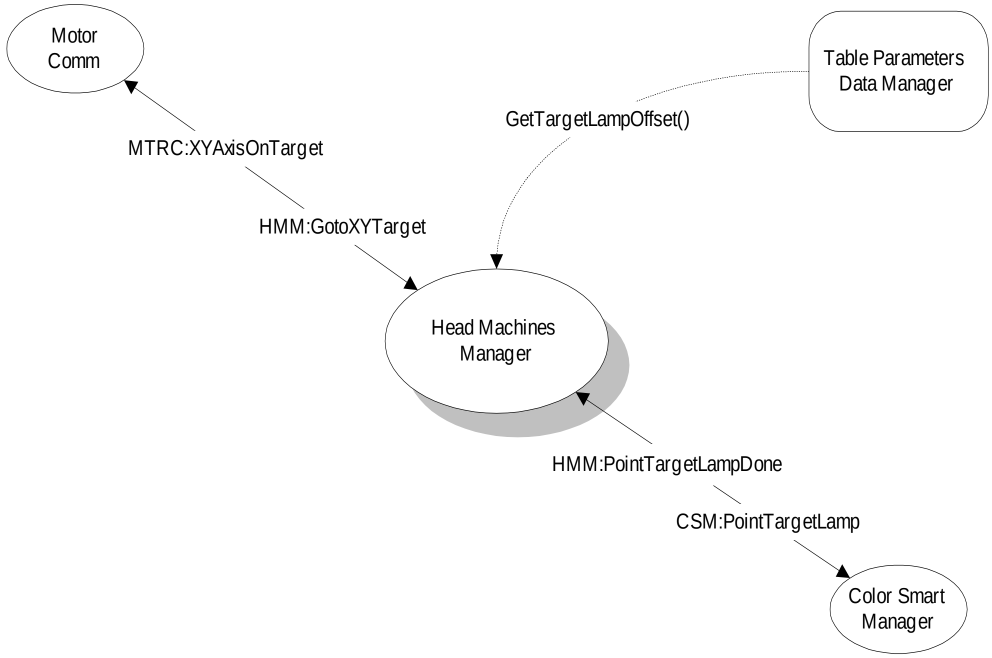


5.13.1.5. HMM System Interaction, Point Target Lamp

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:

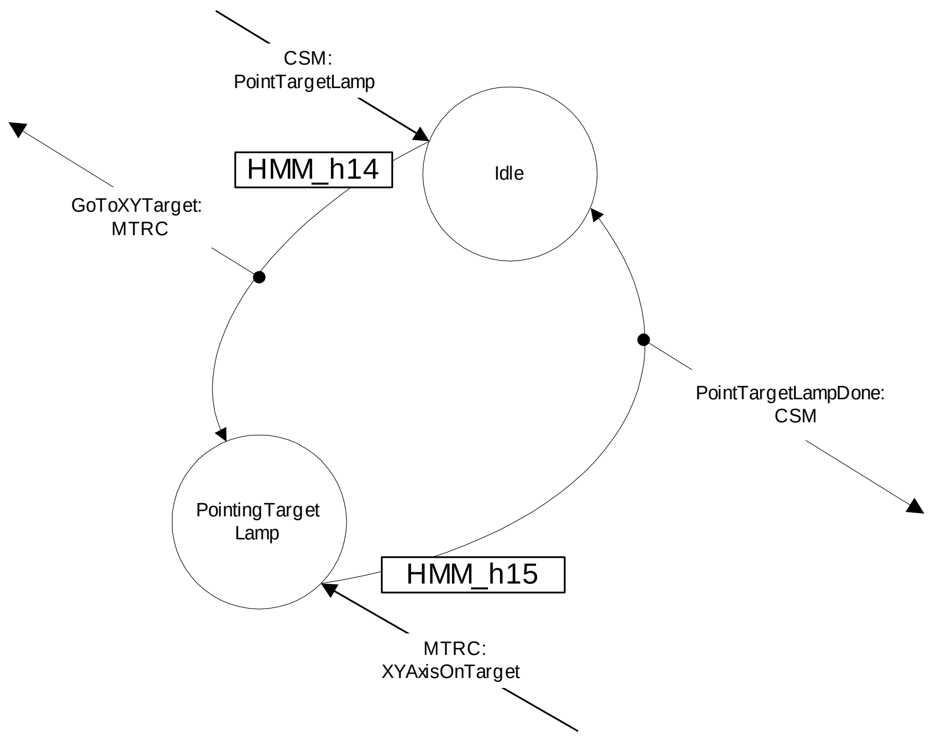
Activating and deactivating the target lamp requires sending another message.

5.13.1.5.1. 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:

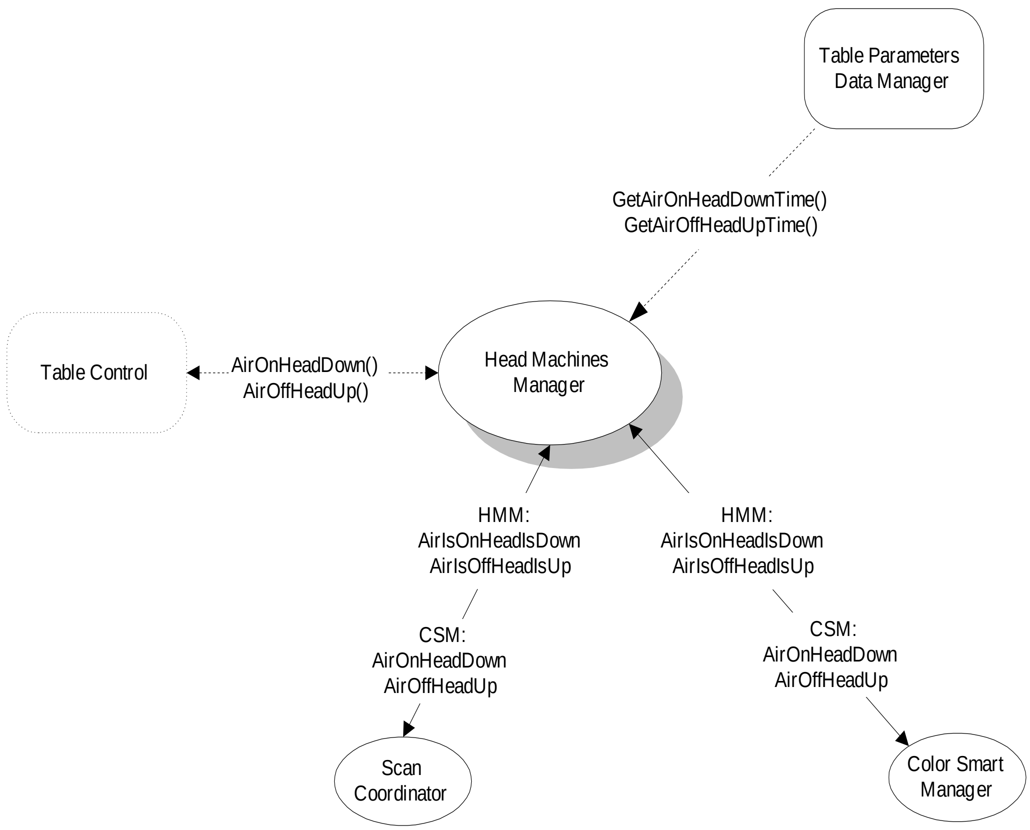
Unlike the “Move To Position” command this process does not have the settle delay. In addition to the “Turn Target Lamp On/Off”, this request is primarily used to show the end-user a specific point within the table or sheet.

5.13.1.6. 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.

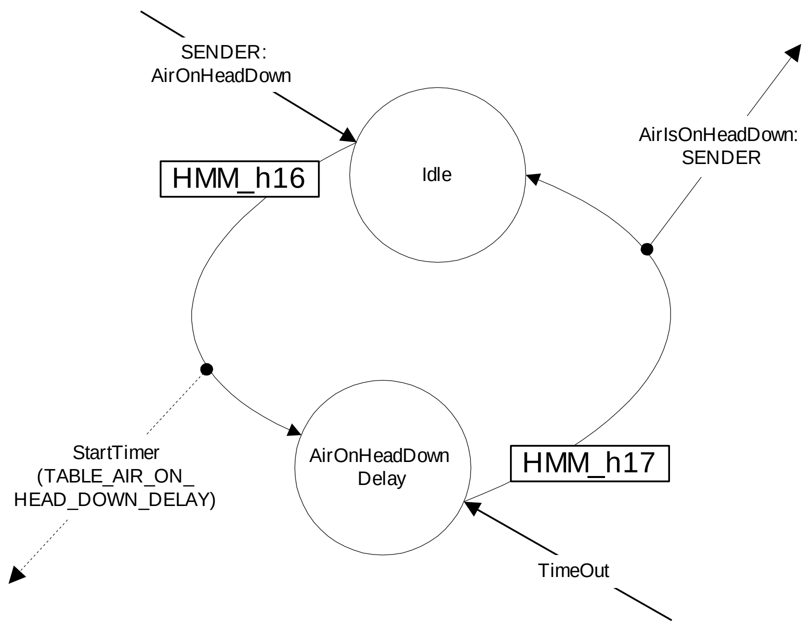


5.13.1.6.1. HMM – Air On Head Down, State Transitions

CSM or HOC generates the “AirOnHeadDown” message.

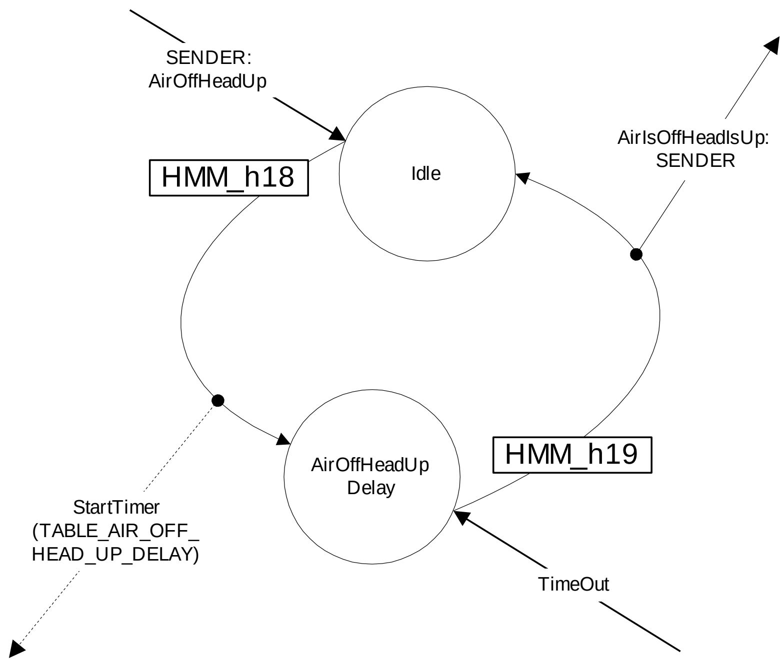
When the message is received by HMM, it activates or deactivates the air using the services of TC. It then performs a “Wait” to ensure that the head is either fully lowered or fully raised before he goes back to the idle state. Once HMM is in the idle state, a measurement command can immediately be processed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

The SENDER is notified about the completion of the request.



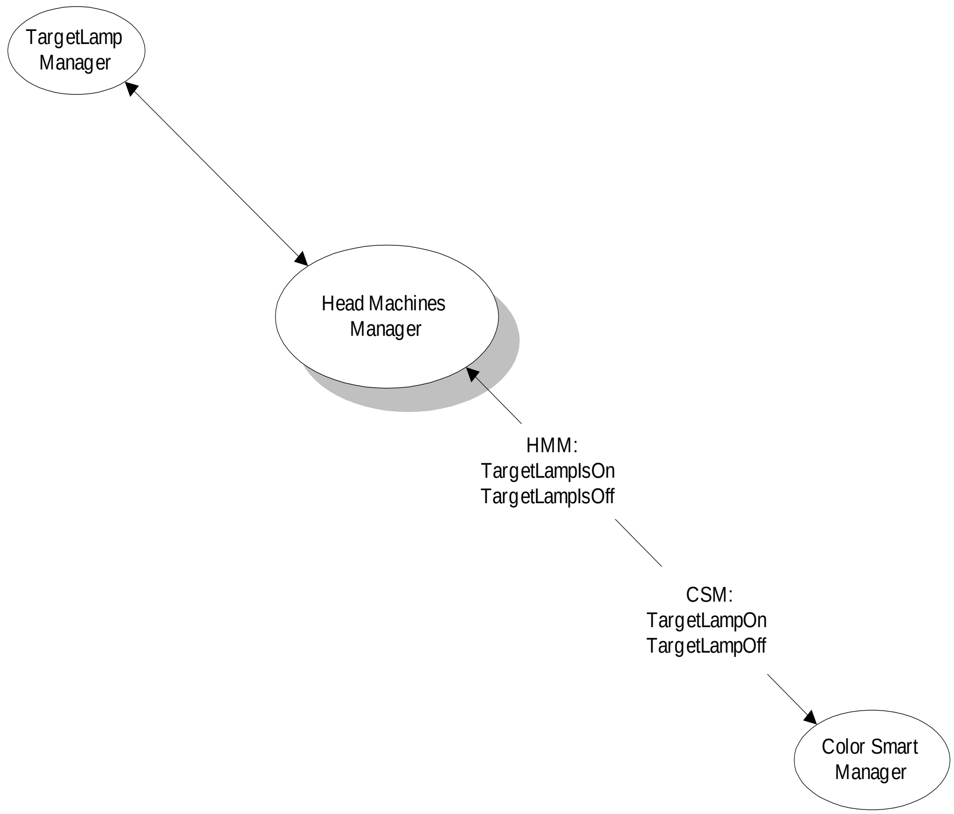
5.13.1.6.2. HMM – Air Off Head Up, State Transitions

CSM or HOC generates the “AirOffHeadUp” message.



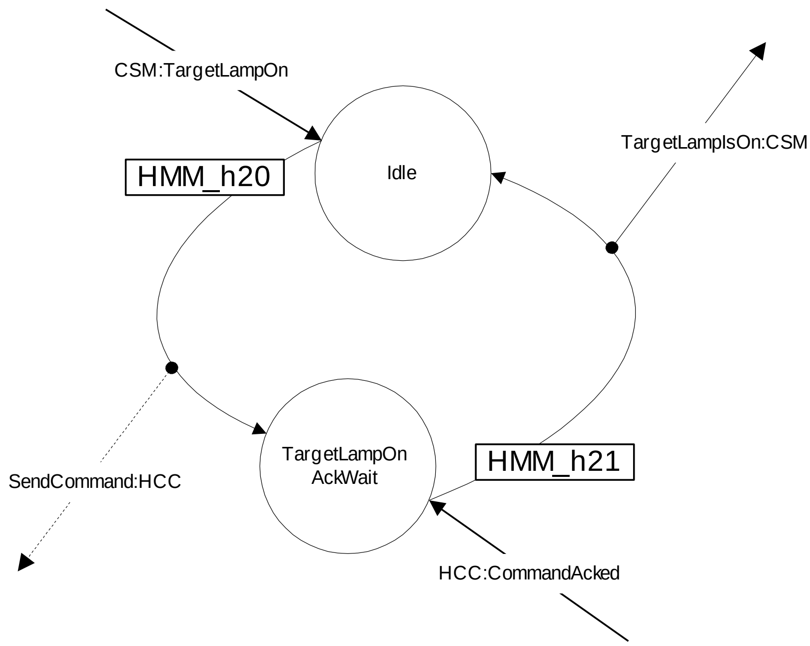
5.13.1.7. HMM – System Interaction, TargetLampOn / TargetLampOff

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



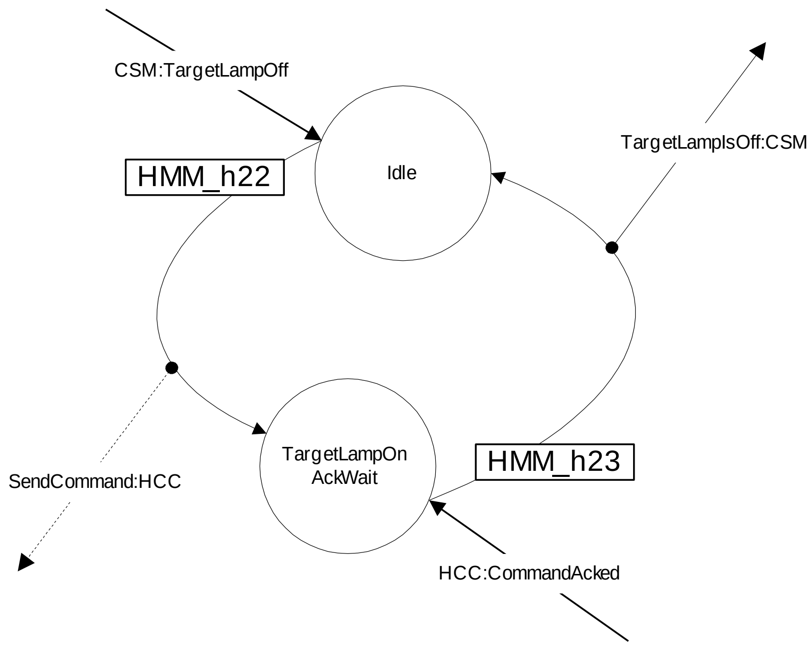
5.13.1.7.1. HMM – Target Lamp On, State Transitions

CSM generates the “TargetLampOn” message.



5.13.1.7.2. HMM – Target Lamp Off, State Transitions

CSM generates the “TargetLampOff” message.

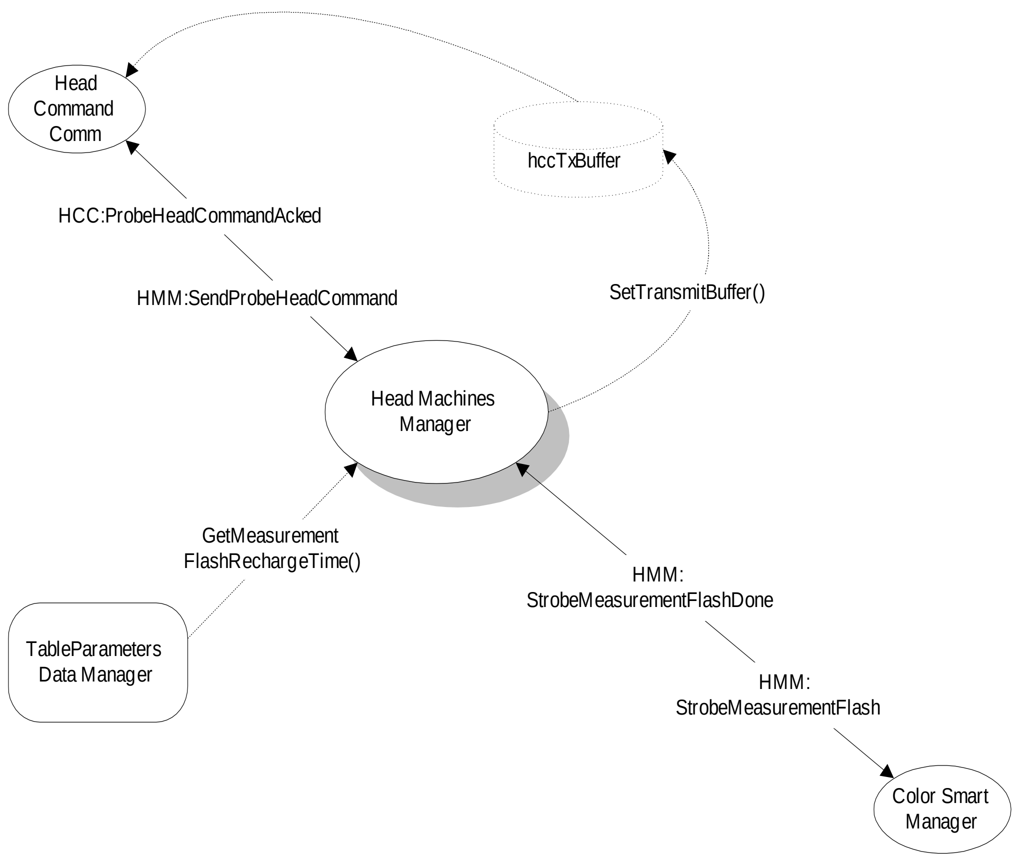


5.13.1.8. HMM – System Interaction, StrobeMeasurementFlash

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.

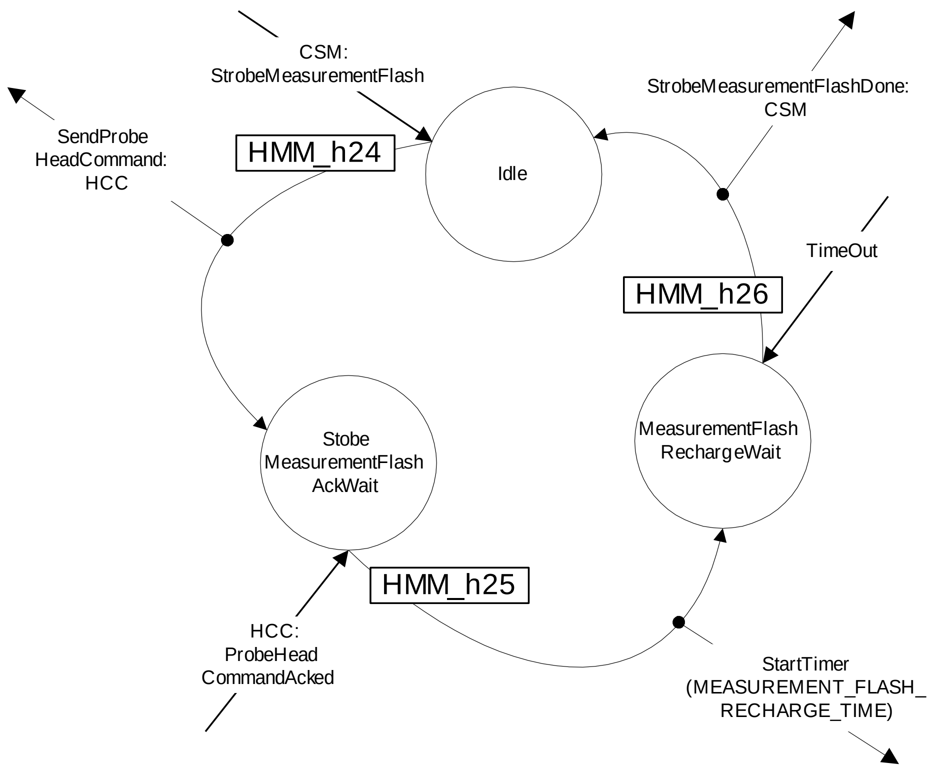
The diagram shows all the machines involved in performing the process.

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



5.13.1.8.1. HMM – Strobe Measurement Flash, State Transitions

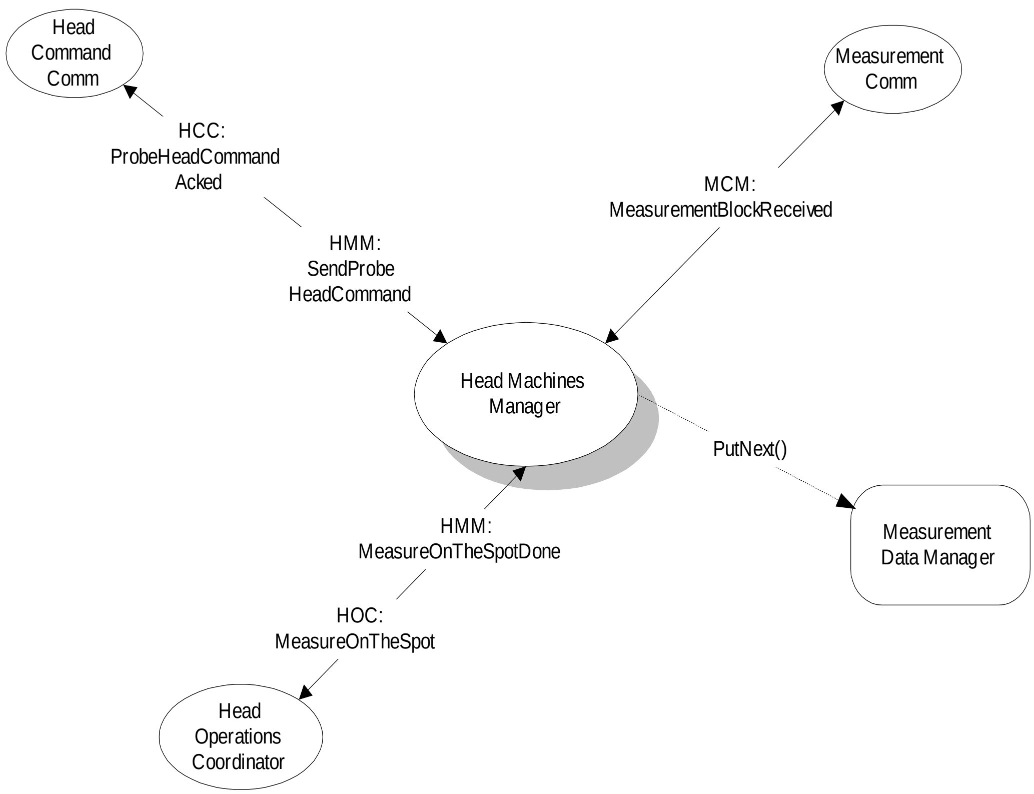
When the message is received by HMM, it fills up the transmit buffer with the probe head command and requests HCC to begin a transmission. HMM is informed once HCC receives the reply (ACK). HMM then performs a “Wait” to ensure that the flash has been given enough time to recharge before he goes back 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.



5.13.1.9. 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.



5.13.1.9.1. HMM – Measure On the Spot, State Transitions

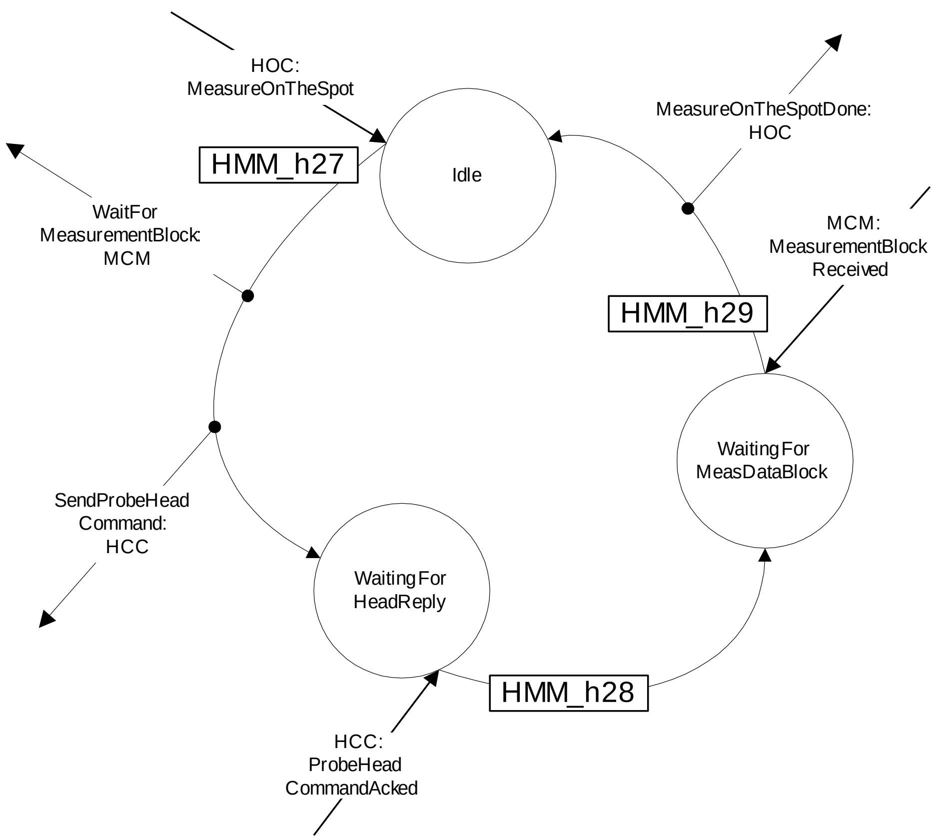
The “MeasureOnTheSpot” message is generated by HOC.

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 49 for more detail.

This process invalidates the Measurement Data Buffer (Shared Memory) as it uses the FIRST slot to store the “Measurement On The Spot” data.



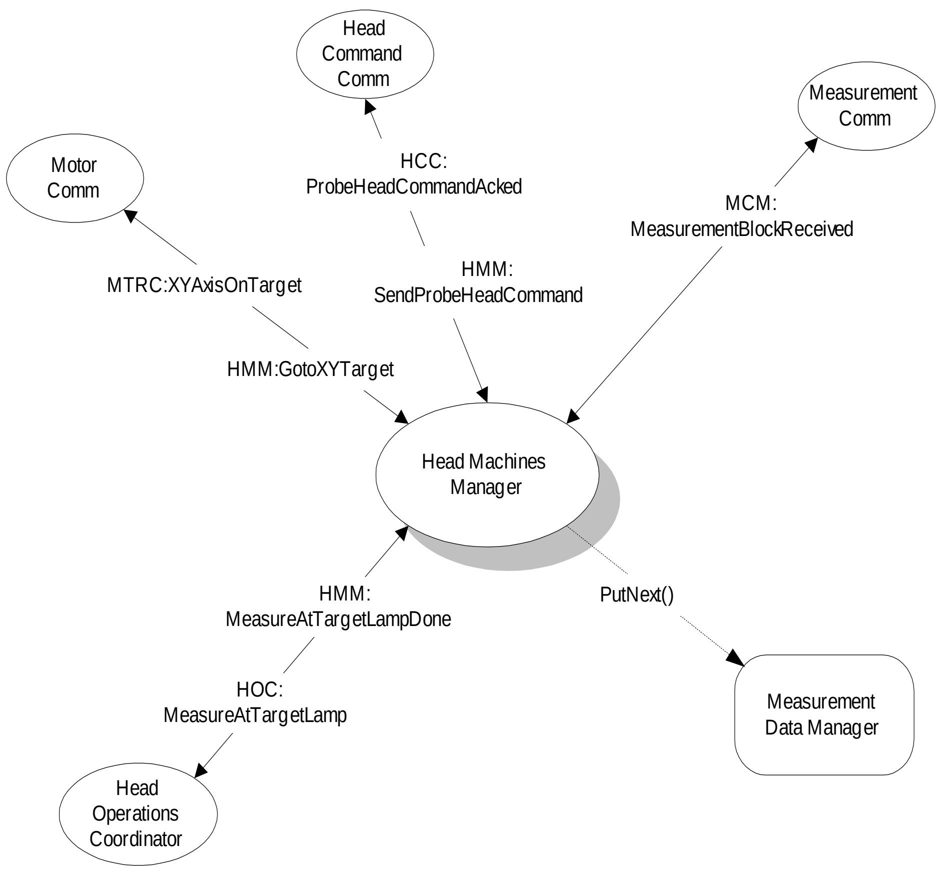
5.13.1.10. HMM – System Interaction, MeasureAtTargetLamp

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.

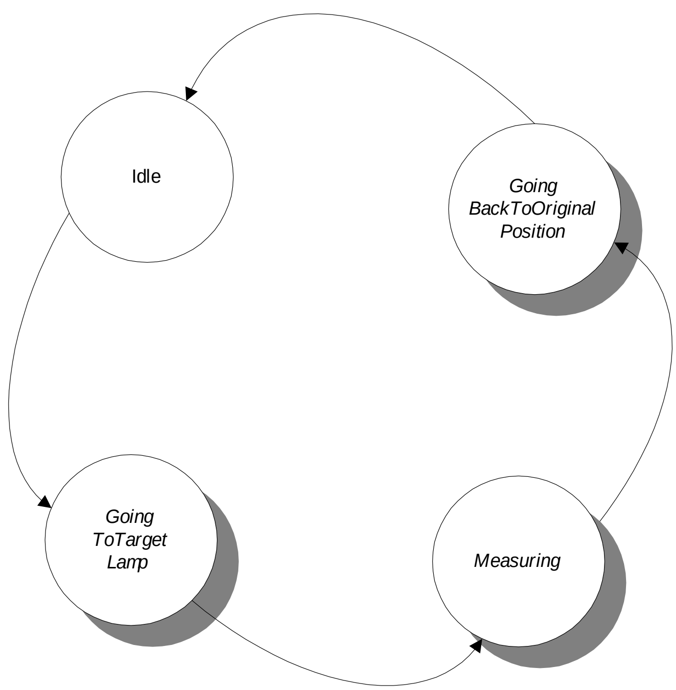


5.13.1.10.1. HMM – Measure At Target Lamp, State Transitions Level 1

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.

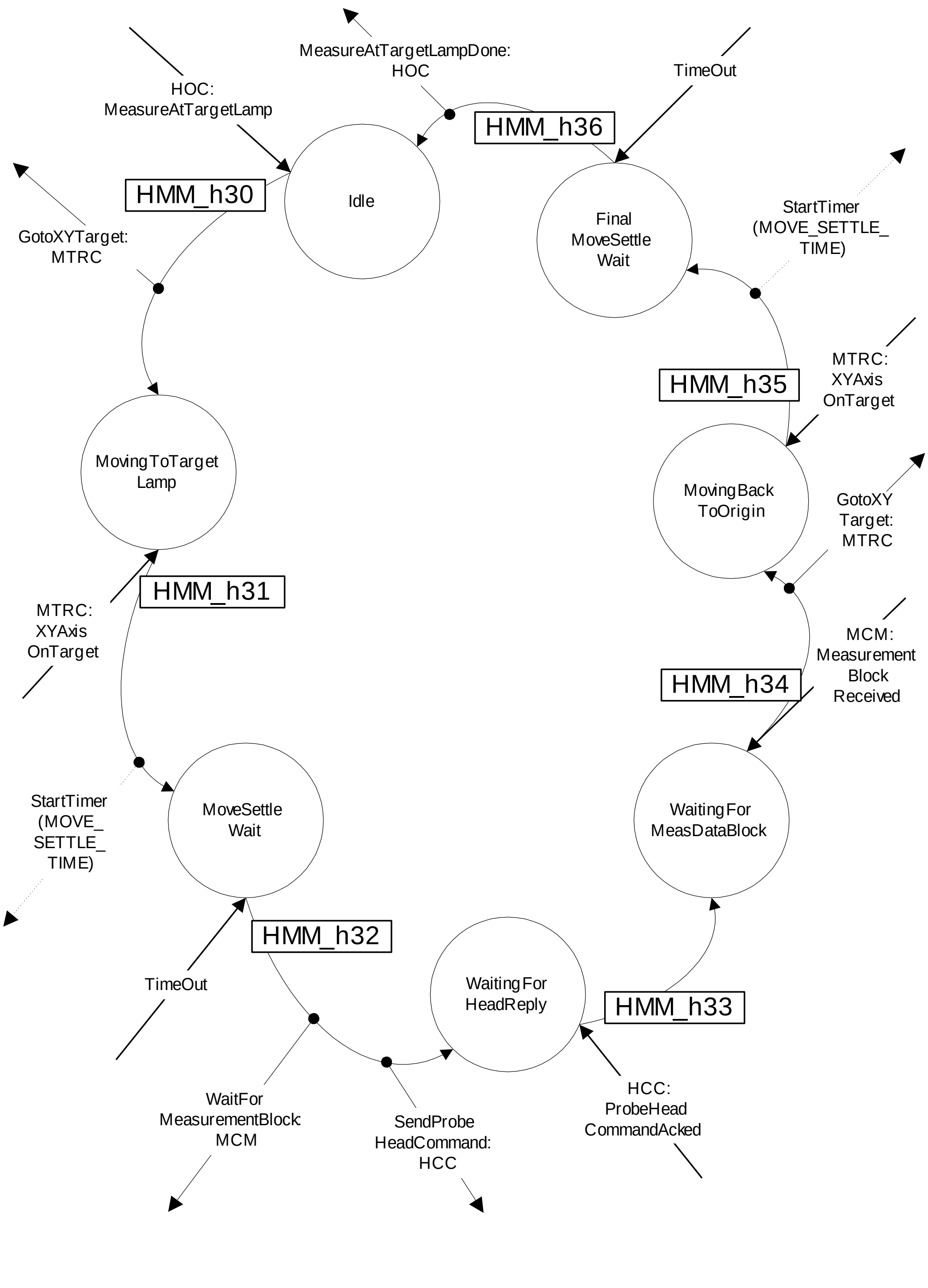


IMPORTANT:

All the important notes as stated in “HMM – Measure On the Spot, State Transitions” on page 109 also apply for this process.

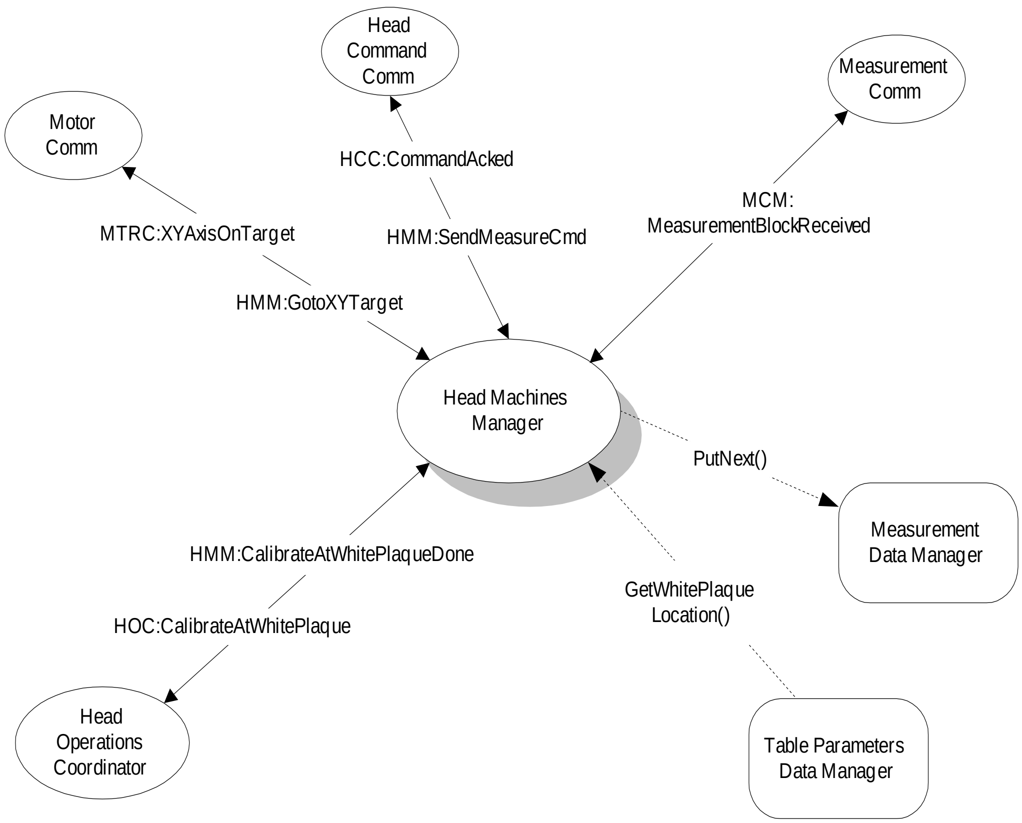
5.13.1.10.2. HMM – Measure At Target Lamp, State Transitions Level 2

This is an expanded view of the “Measure At Target Lamp” behavior.



5.13.1.11. HMM – System Interaction, CalibrateAtWhitePlaque

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



IMPORTANT:

All the important notes as stated in "HMM – Measure On the Spot, State Transitions" on page 109 also apply for this process.

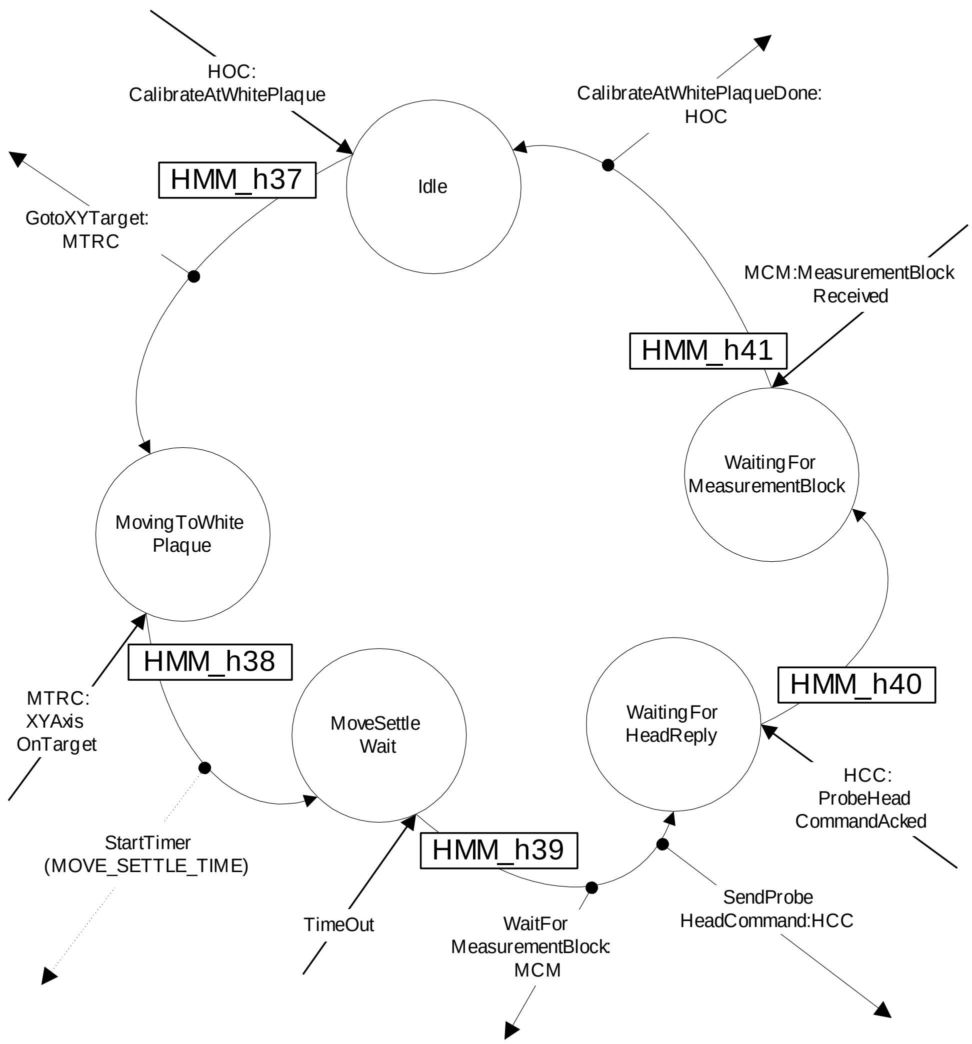
5.13.1.11.1. HMM – Calibrate At White Plaque, State Transitions

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.

HMM knows where the White Plaque is located with reference to the limit switches. 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.

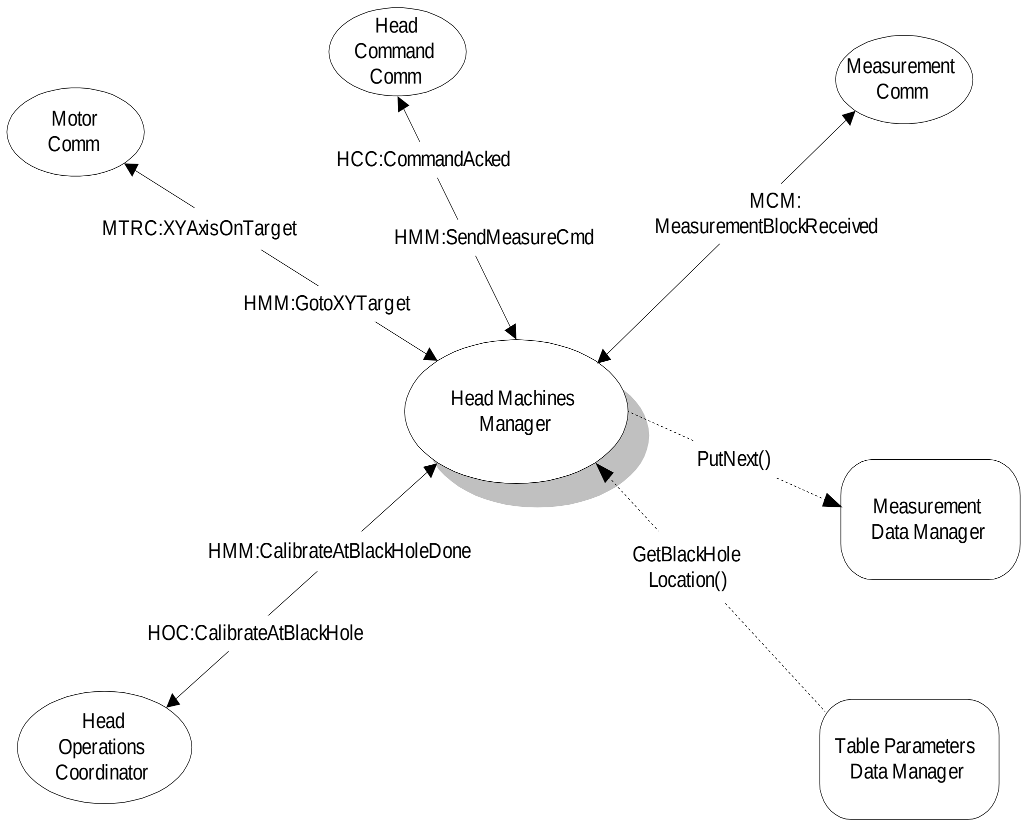
NOTE:

The probe head will be left at the White Plaque after this process is completed.



5.13.1.12. HMM – System Interaction, CalibrateAtBlackHole

The Black Hole calibration is generally done right after the completion of the White Plaque calibration. The diagram shows all the machines involved in performing the process.



IMPORTANT:

All the important notes as stated in "HMM – Measure On the Spot, State Transitions" on page 109 also apply for this process.

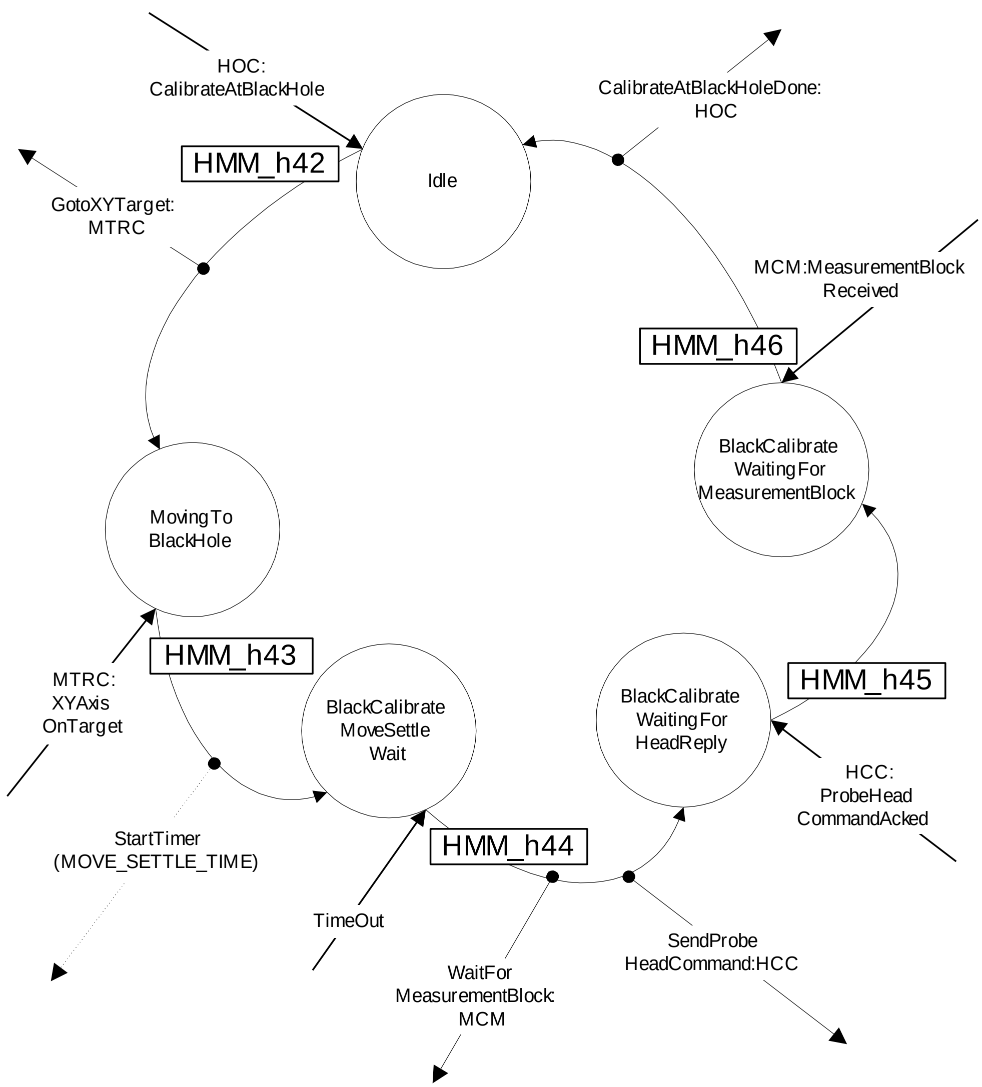
5.13.1.12.1. HMM – Calibrate At Black Hole, State Transitions

HOC generates the "CalibrateAtBlackHole" message.

HMM knows where the Black Hole is located with reference to the home limit switches. This process is exactly the same as that of the White Plaque calibration except for the location of the measurement.

NOTE:

The probe head will be left at the Black Hole after this process is completed.

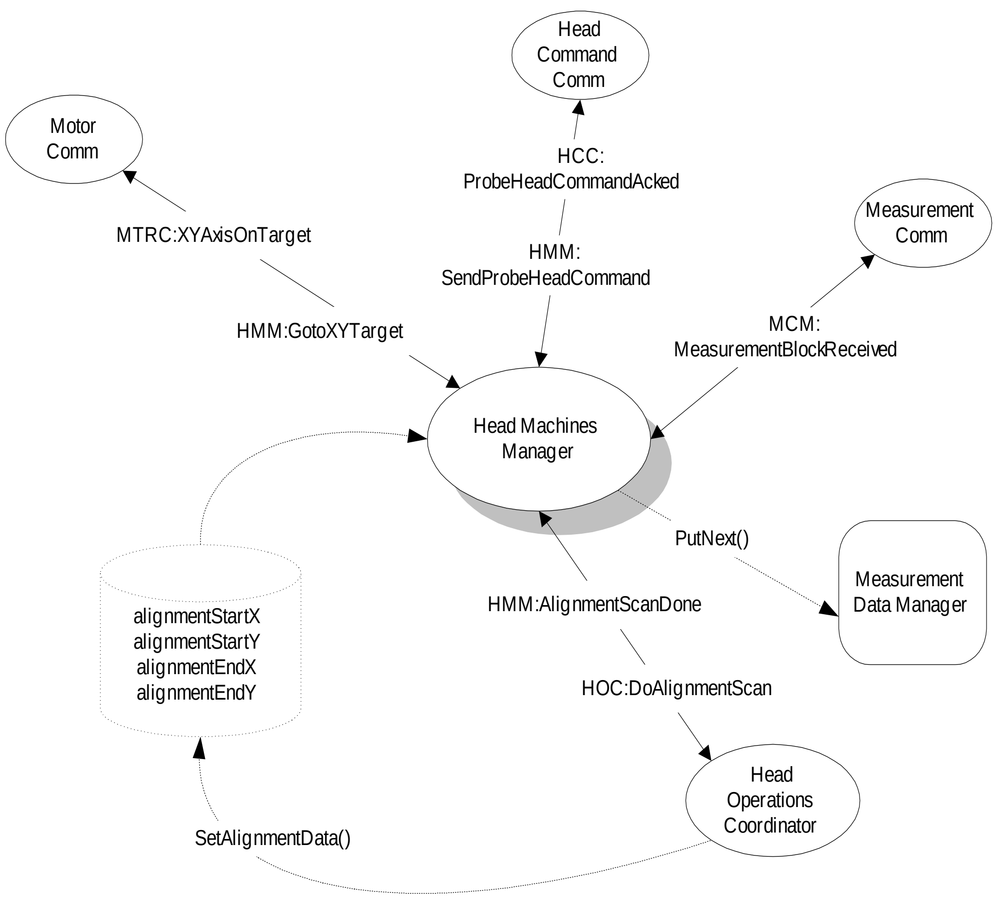


5.13.1.13. HMM – System Interaction, AlignmentScan

IMPORTANT:

In order to provide the highest resolution when searching for alignment point, the motors make 1 step at a time in a given direction. This process of searching also requires precise positioning. Therefore, the backlash compensation mechanism of MTRC will be enabled.

HOC sets the alignment coordinate data and then issues a request to HMM. The diagram below shows all the machines involved in performing the process.



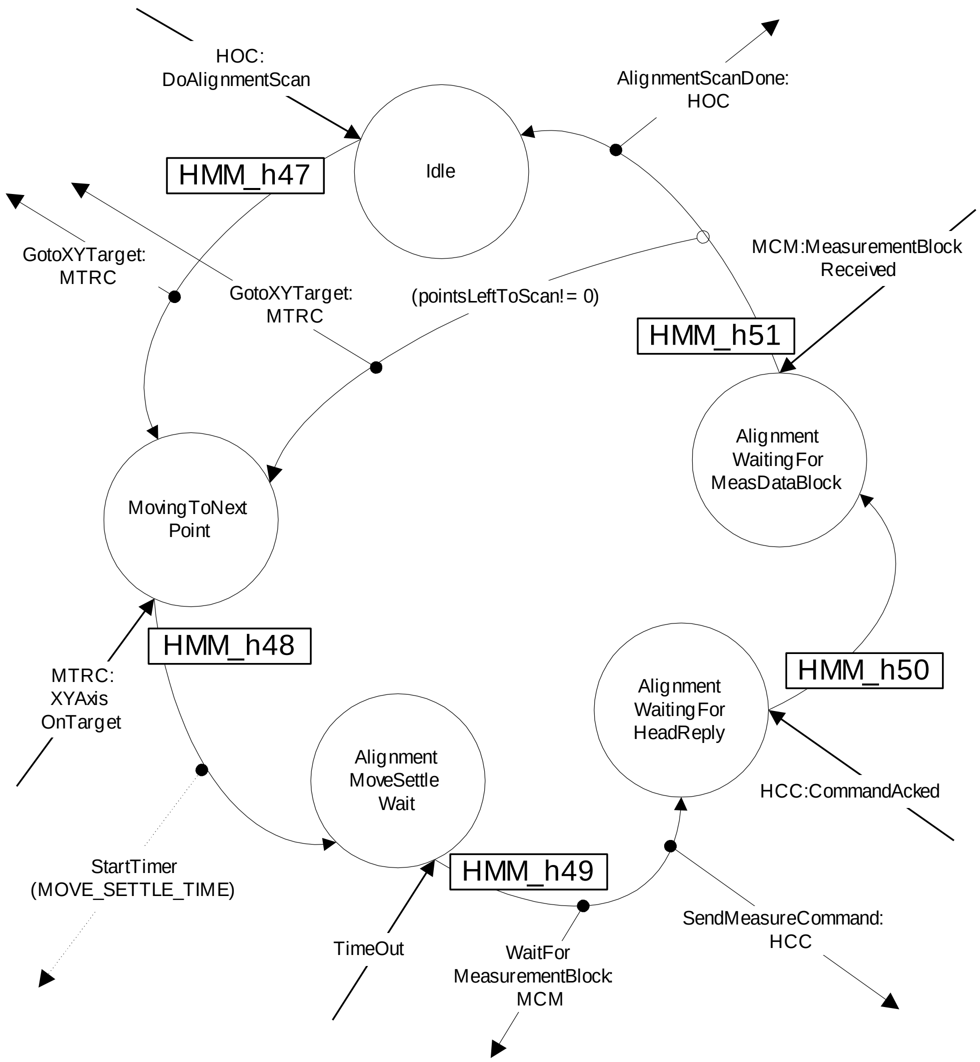
NOTE:

See CSM – (Scanning Mode Idle), Alignment Scan, State Transitions on page 29 for more detail about the alignment scan feature.

5.13.1.13.1. HMM – Alignment Scan, State Transitions

The “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.



IMPORTANT:

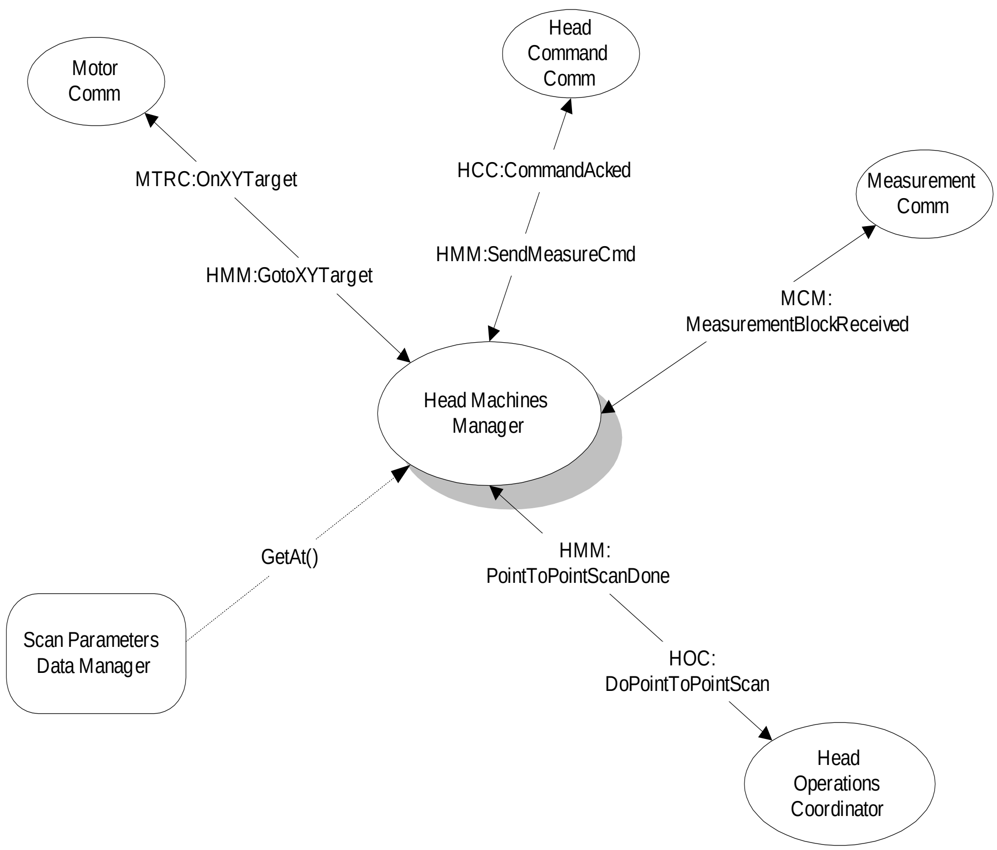
All the important notes as stated in “HMM – Measure On the Spot, State Transitions” on page 109 also apply for this process.

5.13.1.14. HMM – System Interaction, DoPointToPointScan

IMPORTANT:

The "Point to Point" measurement process requires precise positioning. Therefore, the backlash compensation mechanism of MTRC will be enabled.

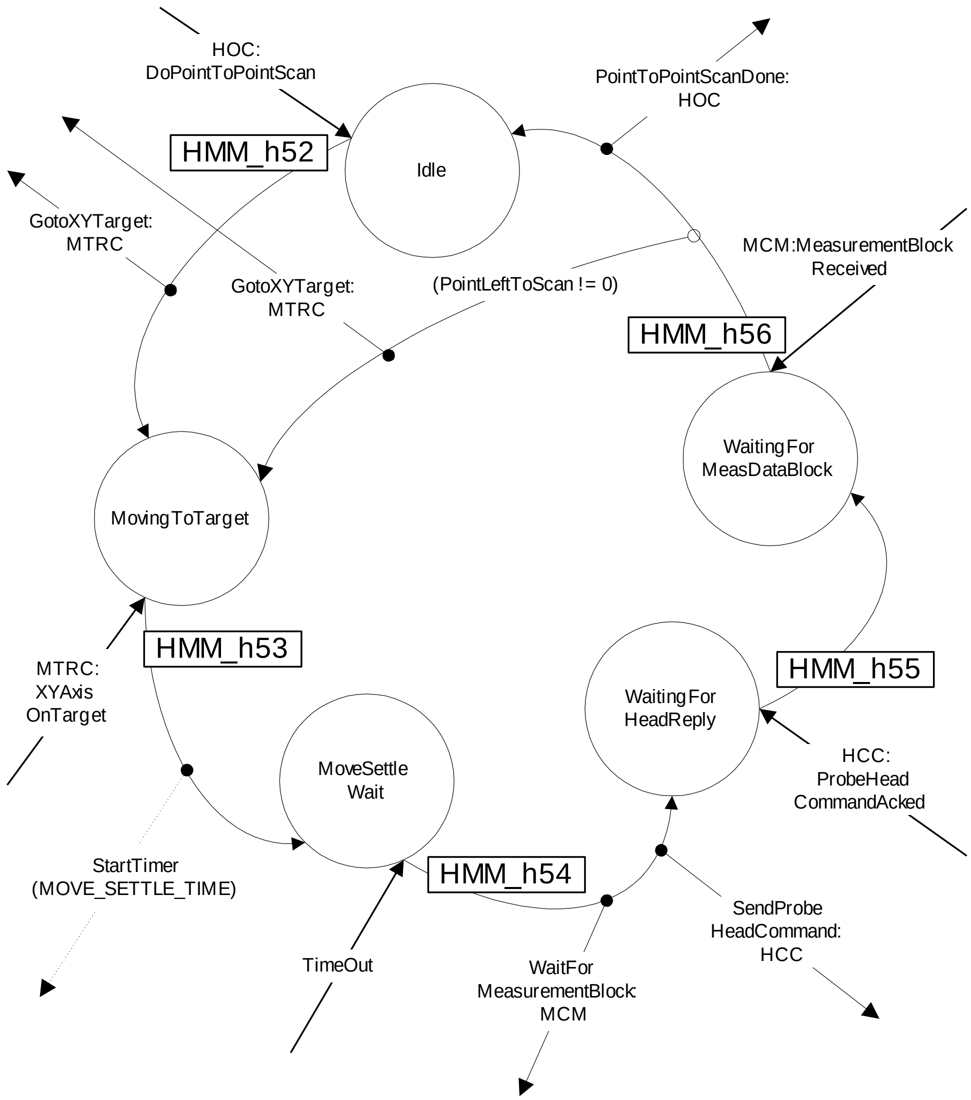
The diagram shows all the machines involved in performing the process.



5.13.1.14.1. HMM – Point To Point Scan, State Transitions

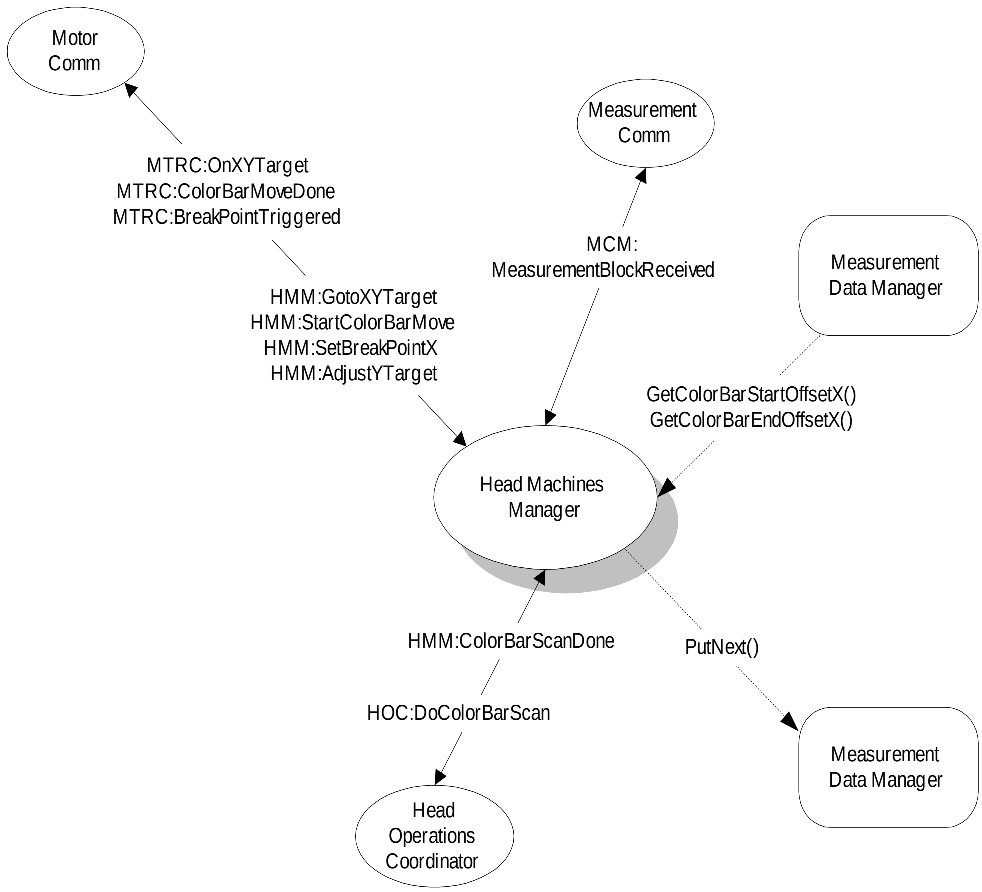
The “DoPointToPointScan” message is generated by HOC.

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



5.13.1.15. 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 are more point measurement data to be expected.

Also see Motor Chip Set Comm (IMCSC) Task Description on page 57 for more information about the Color Bar Scan process.

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

The “DoColorBarScan” message is generated by HOC. This message is accompanied by data.

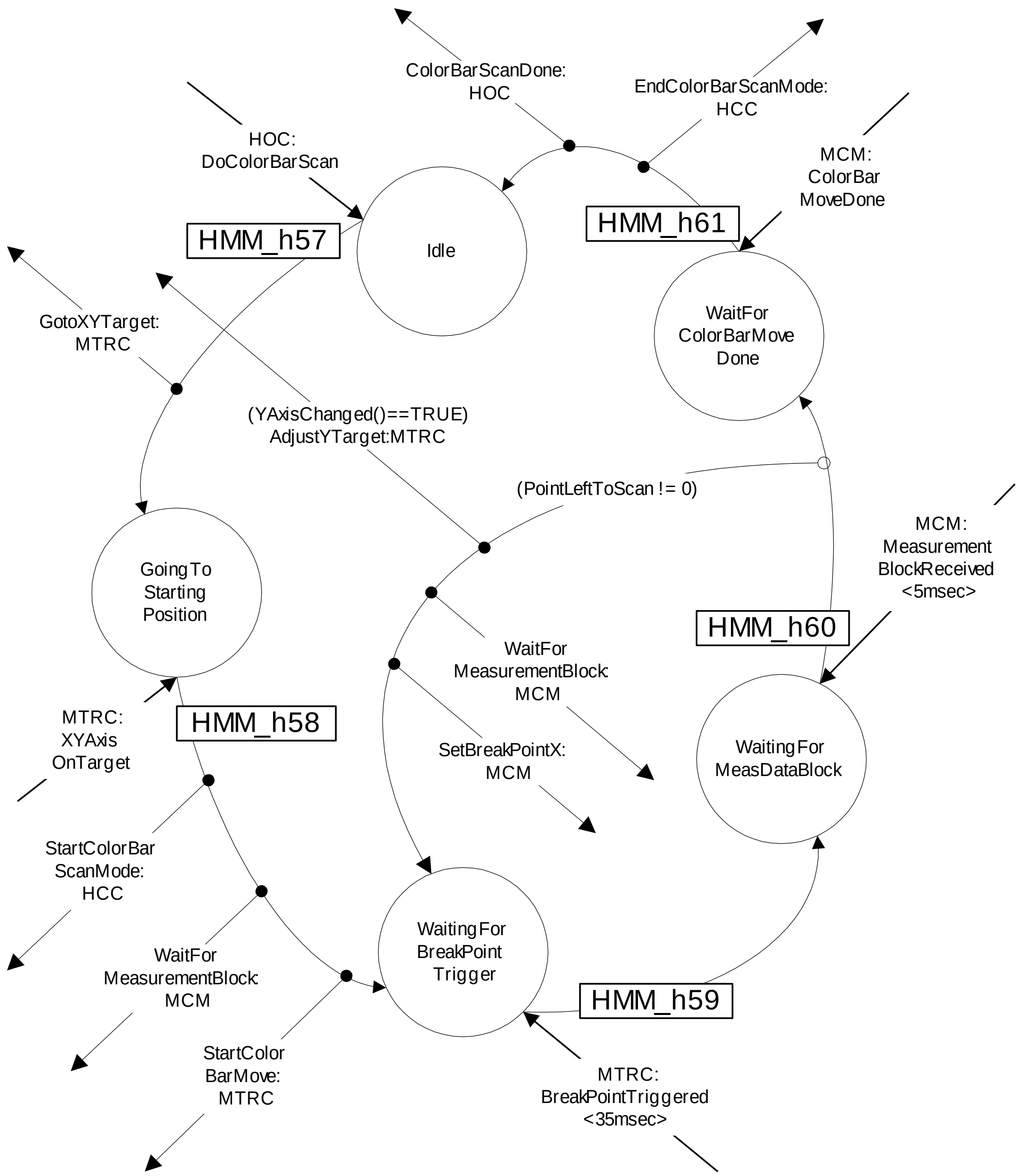
Data1 = scan Start Point Index
Data2 = scan End Point Index

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. “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.

IMPORTANT:

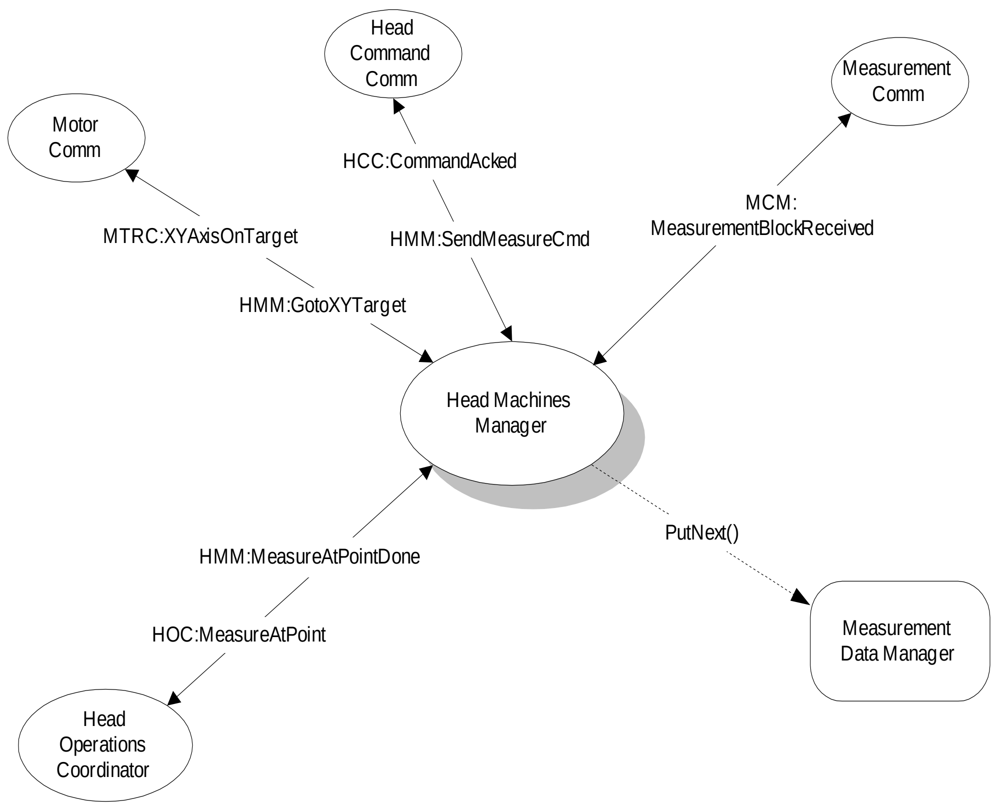
The motor chipset’s break points are updated each time the measurement data block is received. The Y Axis may also be adjusted if necessary.

5.13.1.15.2. HMM – Do Color Bar Scan, State Transitions Level 2



5.13.1.16. HMM – System Interaction, MeasureAtPoint

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



IMPORTANT:

All the important notes as stated in "HMM – Measure On the Spot, State Transitions" on page 109 also apply for this process.

5.13.1.16.1. HMM – Measure At Point, State Transitions

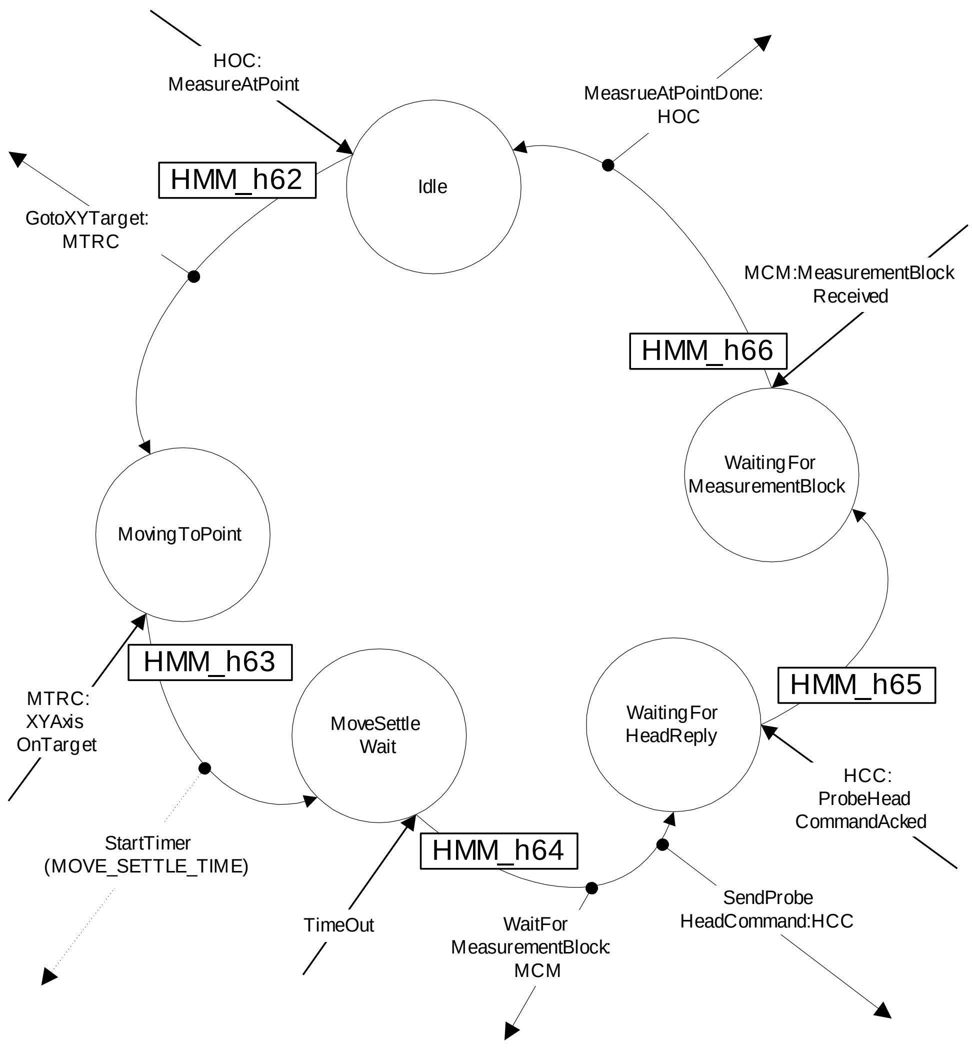
HOC generates the "MeasureAtPoint" message. As stated earlier, HOC is responsible making sure that the probe head is correctly set (Raised or Lowered), before sending this request.

This message is accompanied by data.

Data1 = X Point Coordinate
Data2 = Y Point Coordinate

NOTE:

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



5.14. Status LED (SLED) Task Description

SLED - 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

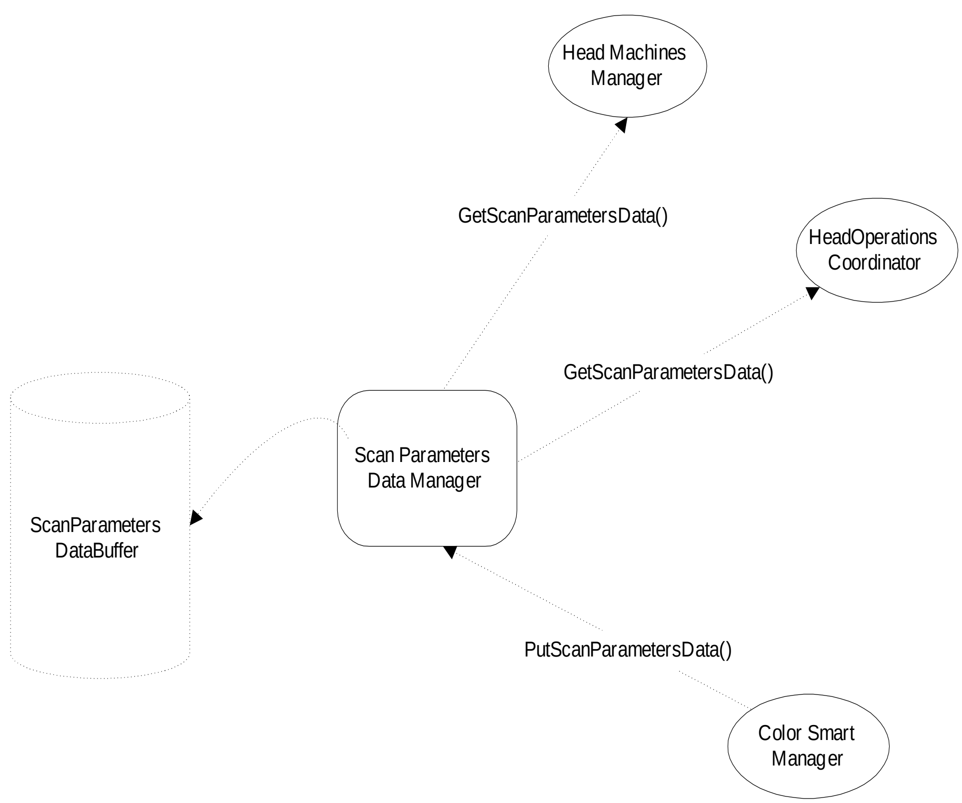
5.15. Scan Parameters Data Manager (SPDM) Task Description

***SPDM - S**can **P**arameters **D**ata **M**anager*

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.

We can have a maximum of 1000 scan coordinate data.



5.16. Table Parameters Data Manager (TPDM) Task Description

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

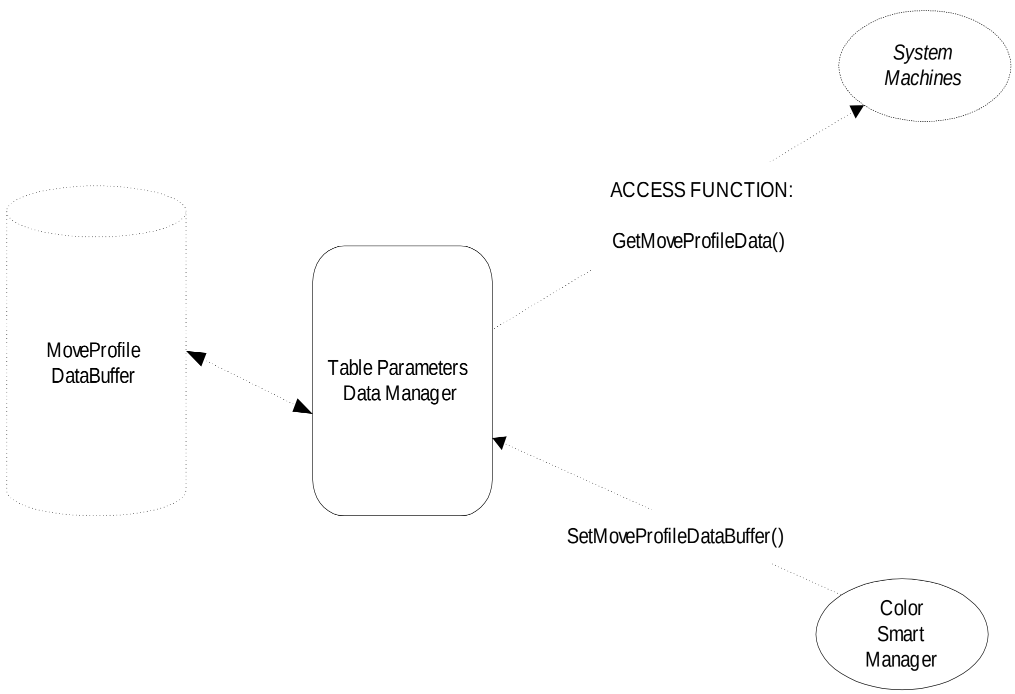
This manager holds all of the table parameters data needed for the CS operation.

IMPORTANT:

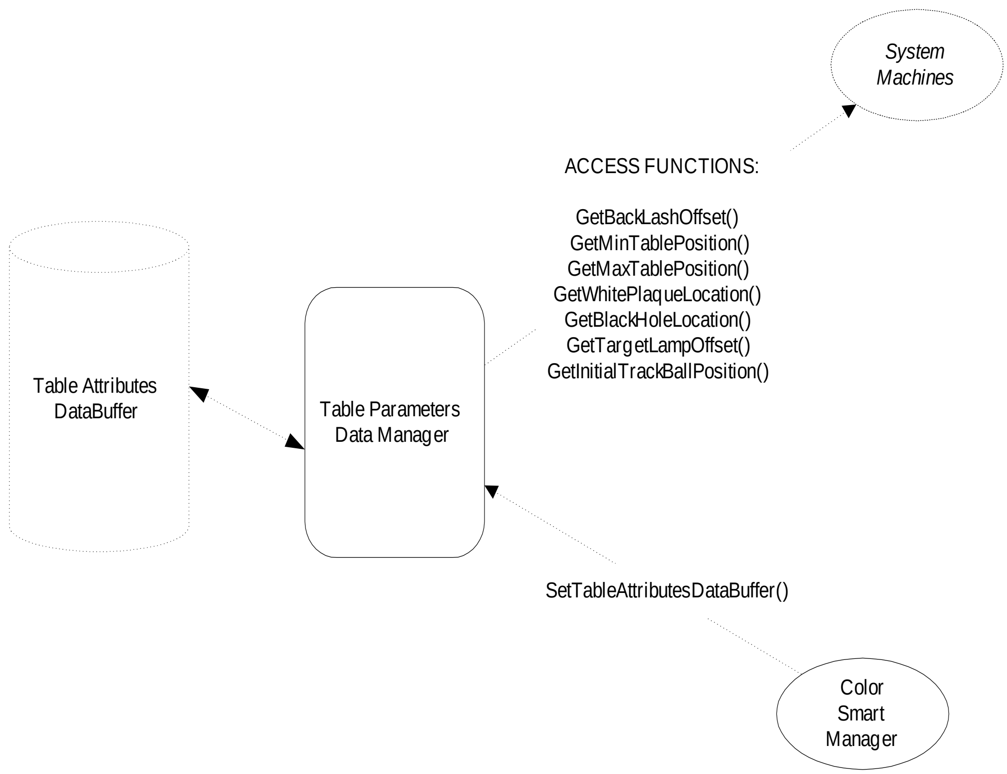
All of the System Machines, who need data from this TPDM, create its own link to this manager. These System Machines make their data request data through the use of the manager's access services.

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".

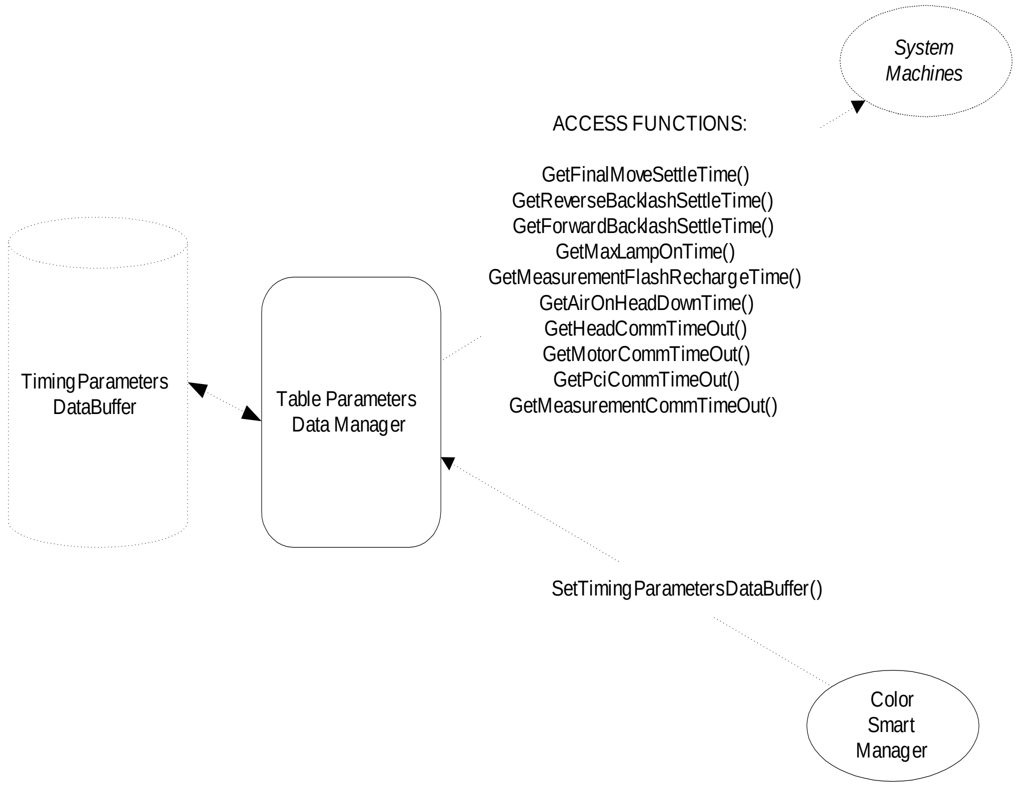
5.16.1. Move Profile Data Access



5.16.2. Table Attributes Data Access



5.16.3. Timing Parameters Data Access

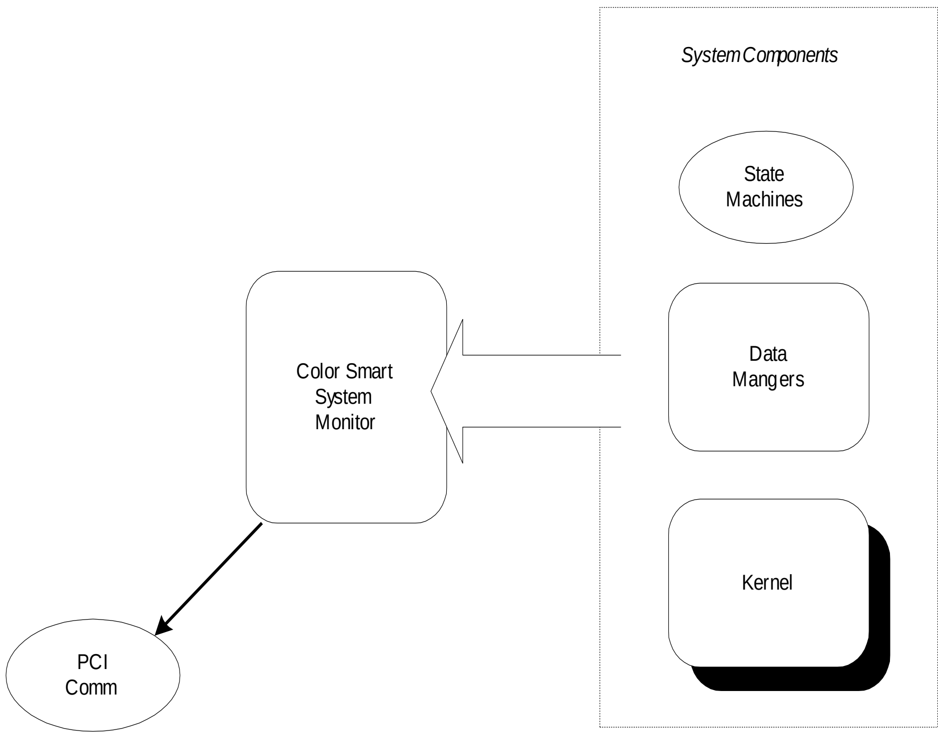


5.17. Color Smart System Monitor (CSSM) Task Description

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.

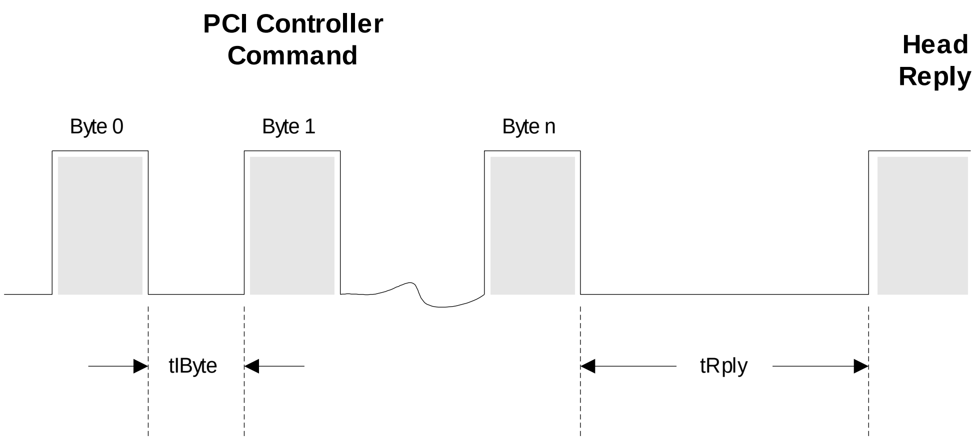


6. PCI Communication Protocol

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

7. Head Communication Protocol

7.1. Head Commands



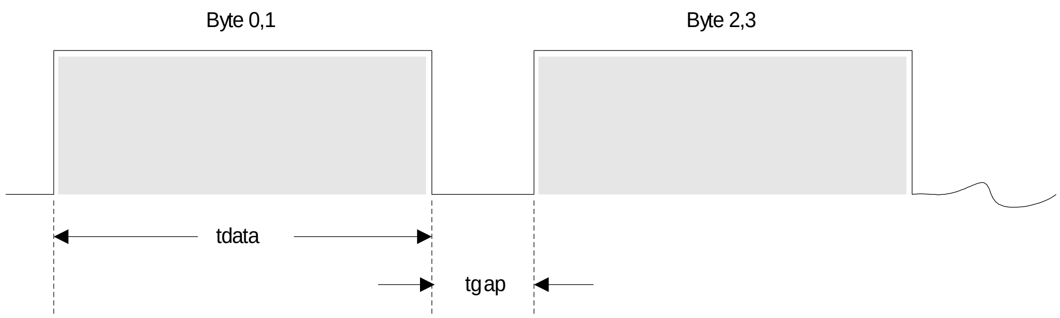
RS232 , 8 Data, 1 Start Bit, 1 Stop Bit, 62.5 KBaud

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

7.2. 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	

8. Hardware

This section contains an overview of the Color Smart Hardware.

8.1. 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.** This can be connected to the external interrupt input called INT4. This port is not used.
IR1	IRQ9	**Unused.** 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.** IR2 is internally dedicated as an input for Timer 1. This peripheral is not used.
IR3	IRQ11	**Unused.** 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.** This can be connected to the external interrupt input called INT6. This port is not used.
IR6	IRQ14	**Unused.** 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.

8.2. 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.

8.2.1. PORT 1 Definition

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

8.2.2. PORT 2 Definition

Port Number	Usage Description
P2.0	**Used as Chip Select 0 (CS0).** CS0 selects one of the banks of the Local System Ram. P2.0 is multiplexed and disabled.
P2.1	**Used as Chip Select 1 (CS1).** CS1 selects one of the banks of the Local System Ram. P2.1 is multiplexed and disabled.
P2.2	**Used as Chip Select 2 (CS2).** CS2 selects one of the banks of the Local System Ram. P2.2 is multiplexed and disabled.
P2.3	**Used as Chip Select 3 (CS3).** CS3 selects one of the banks of the Local System Ram. P2.3 is multiplexed and disabled.
P2.4	**Used as Chip Select 4 (CS4).** CS4 selects the Motor Chipset Controller when data is being passed back and forth to it. P2.4 is multiplexed and disabled.
P2.5	**Used Asynchronous Serial Communication RX.** **Serial Port 0.** This port is used to communicate (receive replies) from the head module. P2.5 is multiplexed and disabled.
P2.6	**Used Asynchronous Serial Communication TX.** **Serial Port 0.** This port is used to communicate (send commands) to the head module. P2.6 is multiplexed and disabled.
P2.7	**General Purpose Output.** Used for PCI Status LED. P2.7 is multiplexed with CTS (Clear To Send) Serial Port 0, which is unused.

8.2.3. PORT 3 Definition

Port Number	Usage Description
P3.0	**General Purpose Output.** Used for Head Comm Status LED. P3.0 multiplexed with Timer 0 OUT and is disabled.
P3.1	**General Purpose Output.** Used for Motor Chipset Status LED. P3.1 multiplexed with Timer 1 OUT and is disabled.
P3.2	**Used as INT0 interrupt input.** 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.** 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.** Pin is multiplexed with INT2.
P3.5	**Unused.** Pin is multiplexed with INT3.
P3.6	**Unused.** Pin is multiplexed with PWRDOWN.
P3.7	**General Purpose Output.** Used for Head Measurement Status LED. Pin is multiplexed with COMCLK, which is disabled.

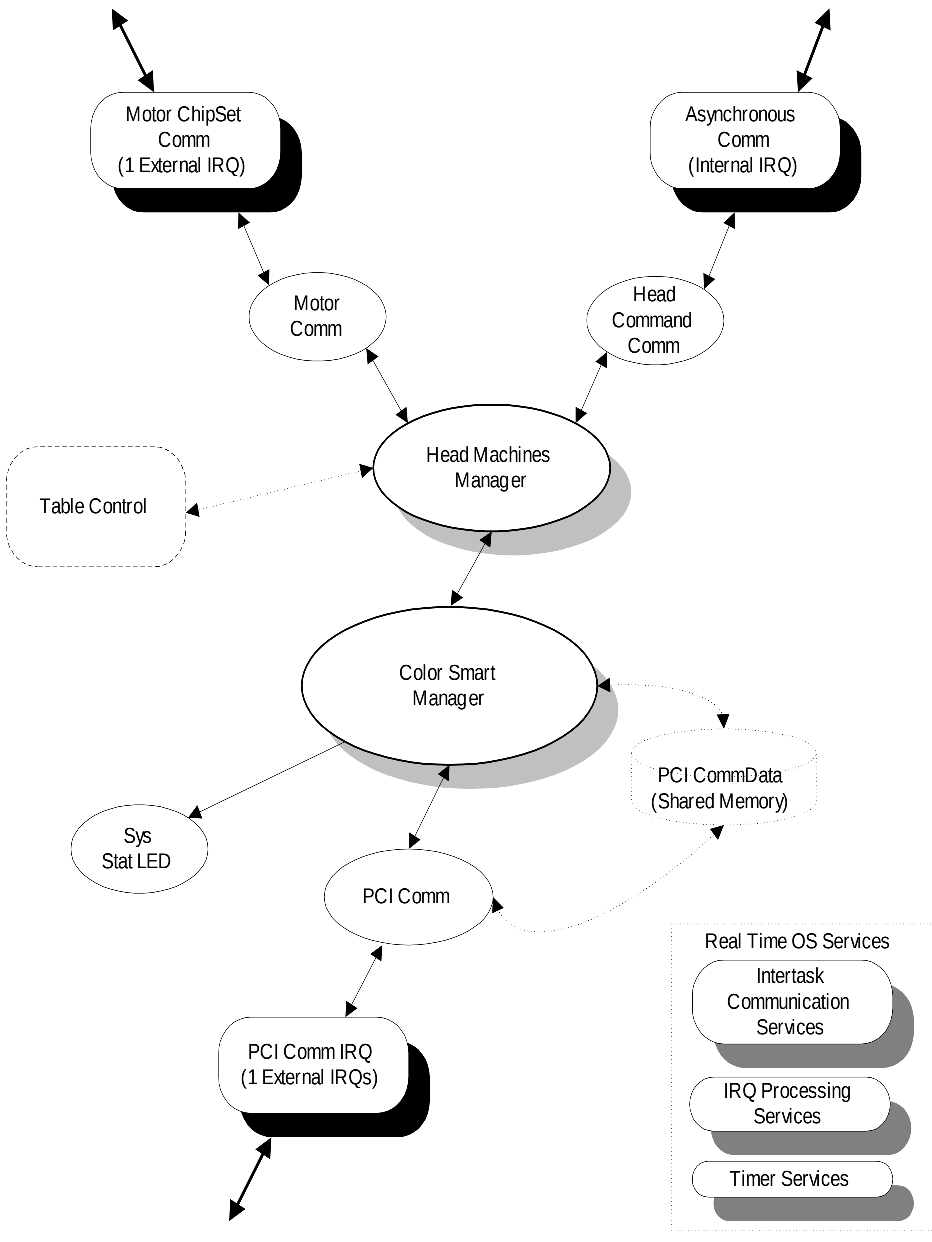
8.3. Chip Select Definitions

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

Chip Select No						
	Mem	IO	Wait States	Bus Size	Range	Description
UCS	**X**		15	8	- 0x3ffffff	64 K BOOT ROM
CS0	**X**		1	16	0x00- 0x1ffff	128K x 16 LOCAL STATIC RAM (BANK 0)
CS1	**X**		1	16	0x20000- 0x3ffff	128K x 16 LOCAL STATIC RAM(BANK 1)
CS2	**X**		1	16	0x40000- 0x5ffff	128K x 16 LOCAL STATIC RAM(BANK 2)
CS3	**X**		1	16	0x60000- 0x7ffff	128K x 16 LOCAL STATIC RAM(BANK 3)
CS4		**X**	2	8	0x300- 0x301	Motor Control Chip Set
CS5	**X**		1	16	0xA0000- 0xA03fff	PCI Mail Box IN
CS6	**X**		1	16	0xB0000- 0xB03fff	PCI Mail Box OUT

9. 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. Memory Maps

10.1. Boot Kernel Memory Map

10.2. Color Smart Main Firmware Memory Map

11. The Color Smart Table

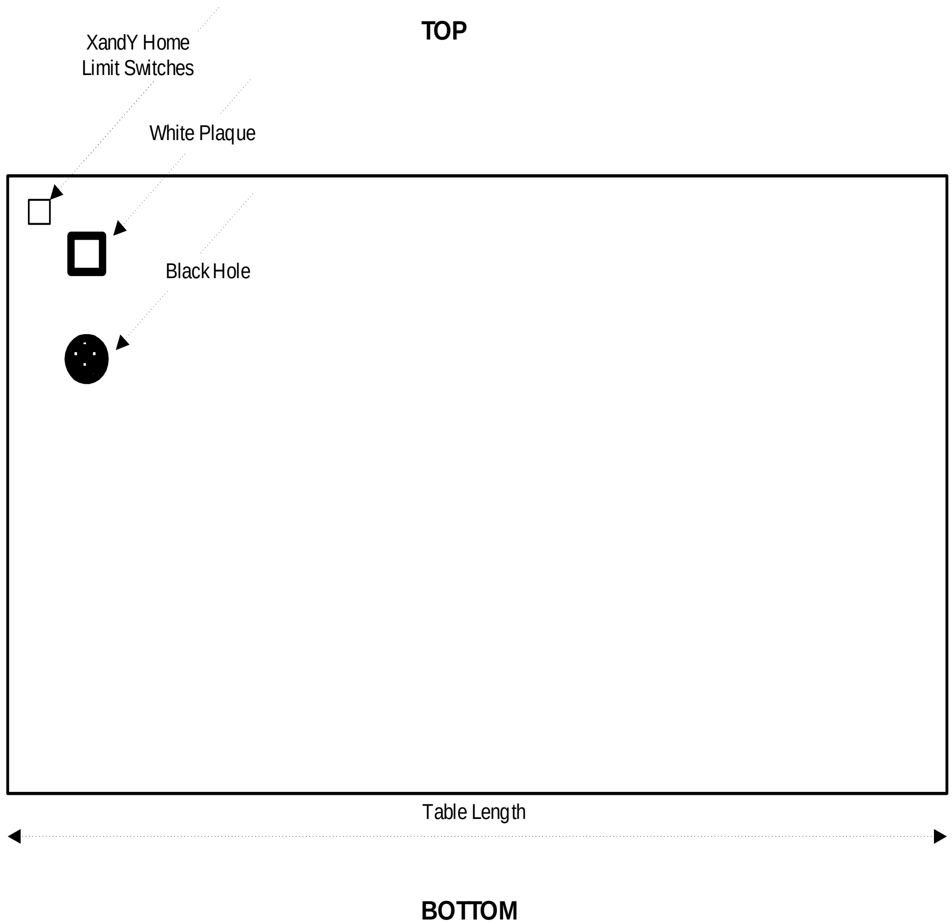
The Color Smart table is similar to the Auto Smart Table (AS10) with only a few differences. Below is a basic description of the table:

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.

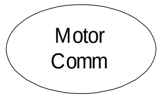
Specific to the Spectral Measurement version is the Black Hole located just below the White Plaque. This is used as a calibration reference as well.

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.

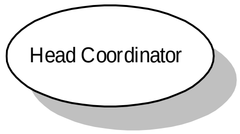
The table comes in a few different lengths depending on the customer's needs.



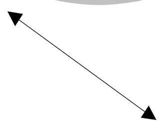
12. Context Diagram Conventions



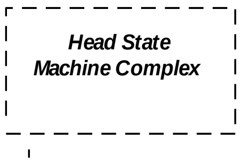
a Finite State Machine (FSM) which handles a specific task. Each of which "capable" of communicating with any other FSM in the systems. FSM can use any of the sevices provided by the OS Kernel.



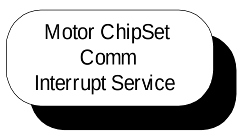
a Coordinator. A Finite State Machine in itself whose task is to handle or manage a set of subordinate state machines.



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.

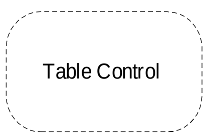


a State Machine Complex comprises of a manager FSM and its subordinates. The label allows special indication for such a dedicated system of FSMs.

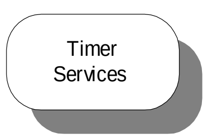


a (Low Level) Driver to indicate the existence of an Interrupt Servicing Routine. Unlike other components of the system, these drivers are initimately coupled to the Hardware Perhiperals as well as the RTOS Kernel.

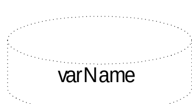
a Data Manager providing access to a component of the system. This accomplishes the "data hiding" aspect of an object oriented system.



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.

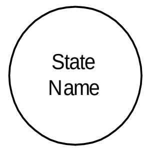


a Data Store. This represents a storage in memory. A variable in short. Can also be a structure or arrays...Connection to the data store indicates how the data is shared among the different objects.

HCC:EnableSComm

a Message "EnableSComm" sent by HCC .

13. State Transition Diagram Conventions



a State Bubble

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

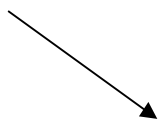
GotoXYTarget:MTRC

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.



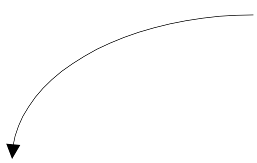
an Incoming Communication line.

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

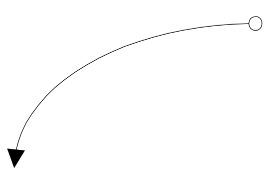
Also known as a Message Handler.
Each state transition begins by excuting an Exit Procedure which performs all of the necessary things required to change the State Machine's state. SMNAME_h1 is the actual label of the exit function as written in the source code.



a State Transition Path Line

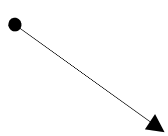
This is a light curved line that begins from one state bubble and ends at another state bubble. This represents the how the states will chang e bas e on the current event or system conditions.

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



a State Transition Branch Path Line

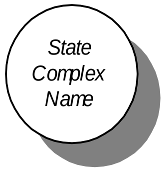
This is the same as a State Transition Path Line except that it begins with a light dot. This Line symbol is used when the state can change to more than one state. This line is usually accompanied with a decision statement.



an Out-Going Communication line.

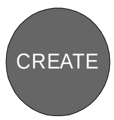
This is a light line that begins with a heavy dot and ends with with an arrow. The arrow points away from a State Transition Path Line. This Communication line is always accompanied with a text indicating the message and the receiver ID

14. 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.