NRC·CNRC
From Discovery to Innovation...

Science at work for Canada

# HERZBERG EXTENSIBLE ADAPTIVE REAL-TIME CONTROLLER (HEART)

# Internal Interface Definition Document

National Research Council Canada
Herzberg Astronomy and Astrophysics
February 2021

# Table of Contents

# 1. INTRODUCTION

## 1.1 BACKGROUND

To ease the understanding of the HEART RTC structure, we use the concept of a “block”, which is a reusable software unit, from which an RTC can be composed. It performs various tasks and a pipeline stitches together blocks to make an RTC.

HEART RTCs have real-time inputs and outputs, but not all inputs and outputs are the same, yet HEART needs to deal with those inputs and needs to know how to create those outputs. There are standardized blocks for each of the inputs and outputs and there are custom handlers to deal any unique interface. For example, an input WFS may need custom code to accept the input, then potentially translate that input to an expected HEART data structure. For outputs there is a standardized block for each type of output that will take the data (e.g. DM command vector), and translate it to be output for the specific hardware. These types of inputs and outputs need to be done at real-time speeds and as a result are tightly integrated with the HEART hard real-time (HRT).

The external interfaces assumes that they are through an Ethernet connection or GigE, but from HEART’s perspective the type of interface with the hardware is handled in the custom handler and could be anything. The custom parts of the interface blocks help with the configuration of the connection.

The primary inputs are pixels and centroids while the primary outputs are Deformable Mirror (DM) commands, Tip/Tilt (TT) commands, Laser Guide Star (LGS) Fast Steering Mirror (FSM) commands and externally streamed telemetry data.

Additional RTC inputs consist of commands and data from the AO system controller. The RTC includes a command handler which processes the commands and data from the AO system controller.

## 1.2 PURPOSE

This document will discuss how HEART handles the internal inputs and outputs. It also includes a description of the example handling functions contained within HEART. The format of commands, data and messages between the RTC and the System Controller are also described here.

## 1.3 SCOPE

This document will only address the custom handlers for inputs and outputs of HEART, also the message format between RTC and AO System Controller.

## 1.4 REFERENCE DOCUMENTS

**RD1** GNAO RTC Interface Definition Document

## 1.5 CHANGE RECORD

| Revision | Date | Section | Modifications |
|---|---|---|---|
| DRF01 | 2020-11-11 | All | JD: Initial draft |

| | | | |
|---|---|---|---|
| DRF02 | 2020-11-16 | All | MS: Draft revision. |
| REL01 | 2020-11-19 | All | Initial Release |
| REL02 | 2021-02-15 | All | CDR Release |

## 1.6 ABBREVIATIONS

**AO –** Adaptive Optics

**DM –** Deformable Mirror

**GNAO** – Gemini North Adaptive Optics

**GVCP –** GigE Vision Control Protocol

**GVSP –** GigE Vision Streaming Protocol

**HEART** – Herzberg Extensible Adaptive Real-time Controller

**HRT –** Hard Real-time

**ICD –** Interface Control Document

**N/A –** Not Applicable

**NGS –** Natural Guide Star

**RTC –** Real-time Controller

**RTS –** Real-time Telemetry Storage

**SRT** – Soft Real Time system

**TBC –** This item still needs to be confirmed

**TBD –** This item still needs to be determined

**TT –** Tip/Tilt

**TTS –** Tip/Tilt Stage

**WC –** Wavefront Corrector

**WFS –** Wavefront Sensor

# 2. SUMMARY INTERFACES

The generic HEART RTC HRT context diagram is shown in Figure 2-1. The RTC Command Handler receives commands and reports status back. Other outputs and inputs are embedded within the HRT, and the External Telemetry Handler part of the storage system.

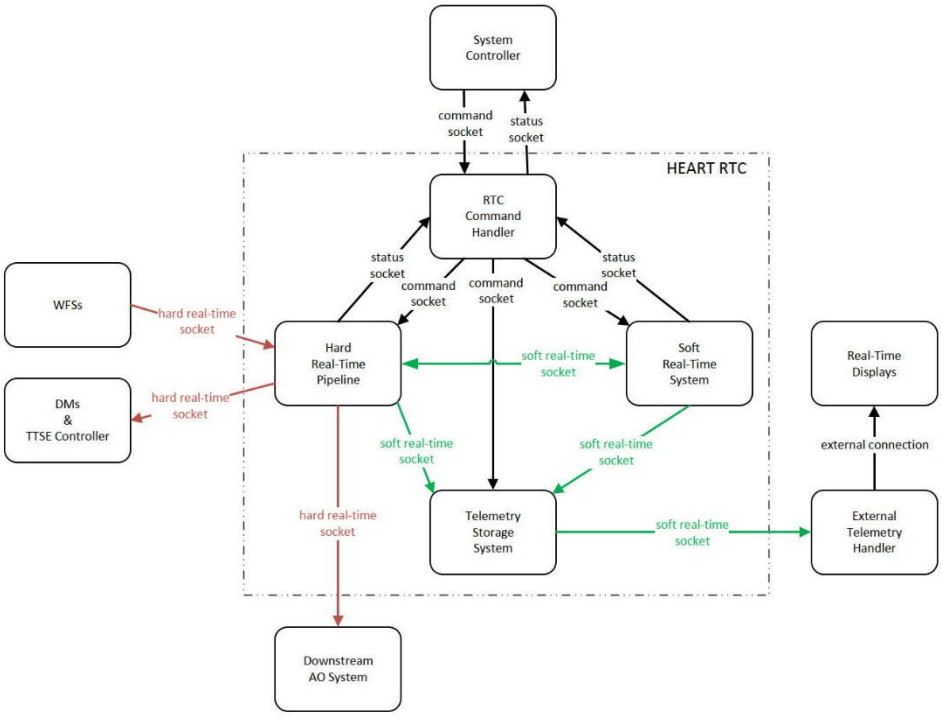



Figure 2-1 - HEART Internal RTC Context Diagram

The types of interfaces included in this document are:

- Configuration file details (Section 3.1)
- Block details along with what pipe they are assigned to (Section 3.2)
- Description and format of circular buffers, including which pipes use those circular buffers. Also further details on the Global Memory circular buffer which contains status information for the various blocks and where the configuration details are saved (Section 3.3)
- Commands to the Blocks (Section 3.4)
- Statistics gathered performed in Blocks for the Soft Real-time (SRT) RPG (Section 3.5)
- Data received in a Block from the SRT-RPG (Section 3.6)
- Enabled/disable flags available at the block level (Section 3.7)

## 2.1 PRIMITIVE DATA TYPES

For reference the different data types noted in this document are:

Table 2-1 - Primitive Data Type Definitions

| Primitive Data Type Name | Number of Bytes | Type |
|---|---|---|
| bool | 1 | boolean value |
| char | 1 | single character |
| byte | 1 | integer number |
| short | 2 | integer number |
| ushort | 2 | unsigned integer number |
| int | 4 | integer number |
| uint | 4 | unsigned integer number |
| long | 8 | integer number |
| ulong | 8 | unsigned integer number |
| enum | 4 | enumerated integer |
| float | 4 | floating point number |
| double | 8 | floating point number |
| string | variable | character array |

## 2.2 MATRIX ELEMENT ORDERING

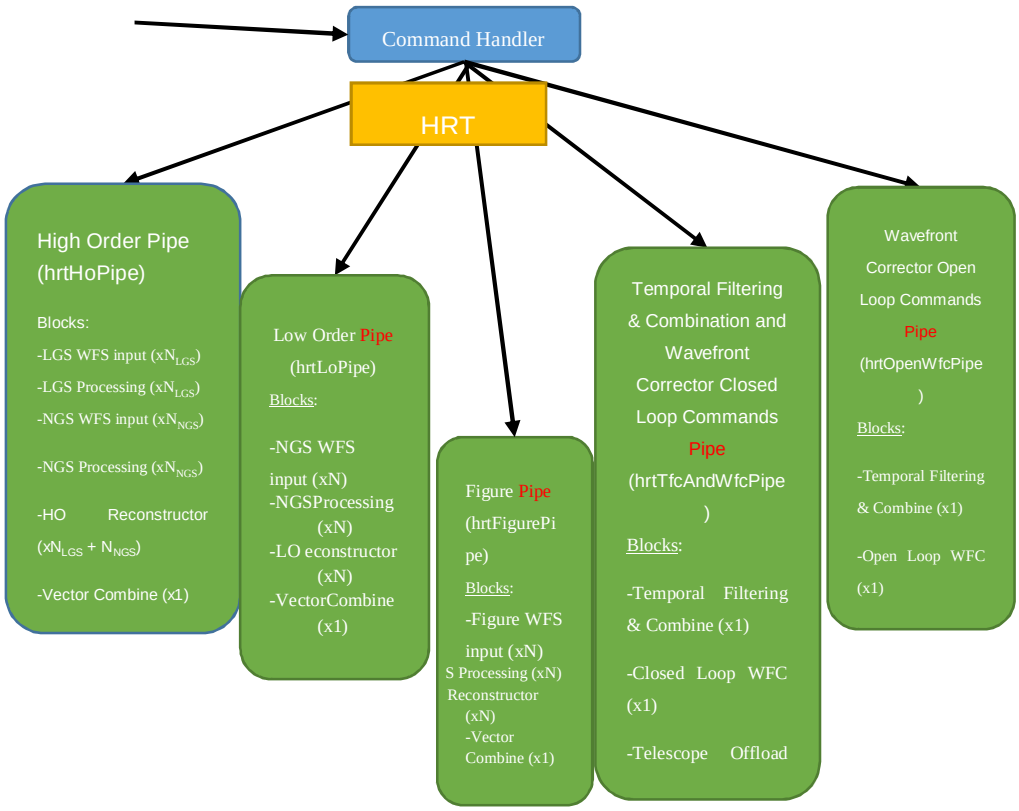
Unless otherwise specified, all matrices and tables transfer to or from the RTC will be in row-major order. Details of how matrices and tables are handled and stored with a sub-system (either the RPG or RTC) are out-of-scope of this document.

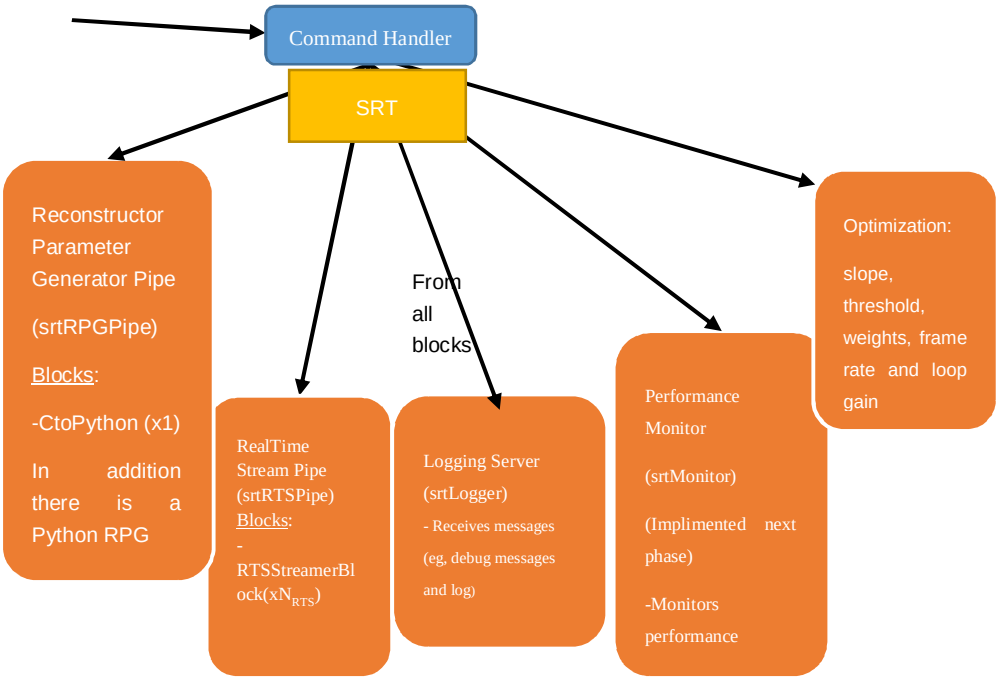
# 3. INTERFACE SPECIFICATIONS

This section details the HEART functions that are internal to HEART such as:

- System controller gets routed through the Custom Interface, then to Command Handler
- Pipes start/stop/send commands to a set of blocks
- There is a client command handler to send commands directly to Command Handler, to the pipelines or to the Blocks directly
- Engineering Interface allows standard commands directly to the blocks
- Standard commands go to the pipe
- Blocks have their own list of custom commands

Figure 3-1- HRT and SRT Pipeline configurations






## 3.1 BLOCK CONFIGURATION (CONFIGURATION FILE CONTENTS)

Each block will have configuration and initialization. This provides information that would be required by the Custom Handler Interface. That information is defined in the specific interface ICD. The following are the types of information that may be considered:

- configuration variable to specify the checksum type within the data packets. This would be added to the appropriate ICDs
- network type order indicating if the data streams are going to be converted to network byte order
- data dimensions

The details of the configuration file are shown in Table 3-1**Error! Reference source not found.**, along with the Block or Pipeline or Command Handler that the information is used in. Please note that additional items can be added when customization is done.

Table 3-1 - Block/Pipeline Configuration Items

| Name | Description | Block(s)/Pipeline |
|---|---|---|
| HO_WFS_COUNT | Number of LGS WFSs | LGS Processing |
| HO_WFS_SIZE | Size of WFSs | LGS Processing |
| HO_WFS_SA | Number of HO sub-apertures per WFS | LGS Processing, HO Reconstruction |
| HO_WFS_GRADS | Number of HO sub-apertures gradients per WFS (equal to 2 *HO_WFS_SA) | LGS Processing, HO Reconstruction |
| HO_WFS_CHUNKS | Amount of HO WFS pixel data chunks to be read before passing on for processing | LGS Processing |

| | | |
|---|---|---|
| SUB_AP_MASK | Sub-aperture Mask (LGS) | LGS Processing |
| SUB_AP_COORD | Sub-aperture X,Y coordinate map (LGS) | LGS Processing |
| I_GRAD_COEEF | Initial Gradient Coefficients for all WFS | LGS Processing |
| LO_WFS_COUNT | Number of NGS WFSs | NGS Processing |
| LO_WFS_SIZE | Size of WFSs | NGS Processing |
| LO_WFS_GRADS | Number of LO sub-apertures per WFS, and number of slopes | NGS Processing, LO Reconstruction |
| LO_NUM_MODES | Amount of LO WFS pixel data chunks to be read before passing on for processing | LO Reconstruction |
| DM_COUNT | Number of modes | HO & LO Reconstructor |
| DM_ACTIVE | Number of active DM actuators | CL WFC |
| DM_SIZE | Total number of actual DM actuators | HO & LO Reconstructor |
| FSM_SIZE | Size of fast steering mirror command | LGS Processing |
| FSM_VALID | Bit mask of enabled FSM | LGS Processing |
| TEMP_FILTER | Temporal filter parameters | Temporal Filtering |
| RPG_PARM | Reconstruction parameters | SRT-RPG |
| LOOP_RATES | Loop rate ranges, HO and LO (Hz) | Temporal Filtering and Combo |
| CPU_ALLOC | CPU core and memory allocations | Pipeline |
| TT_PARAM | Ground layer tip/tilt decomposition parameters | Closed Loop Wavefront Correcting |
| CH_IN_STREAM_IP | IP address and port used by the Command Handler to receive internally streamed data (e.g., from hrtStreamBlocks) | Command Handler |
| MAX_ALT | Maximum layer altitude | SRT-RPG |
| ATMOS_ALT | Atmosphere layer altitudes | SRT-RPG |
| DARK_FILE | Path/filename in a FITS file format containing extensions for first all HO WFS, then all LO WFS | SRT-RPG |
| FLAT_FILE | Path/filename to a FITS file containing extensions for first all HO WFS, then all LO WFS | Temporal Filtering and Combo |
| REF_OFFS_FILE | Path/filename to a file containing reference offsets. Text file, first line contains # off offsets, following lines contains the offsets | LGS/NGS Processing |
| WFC_LAB_FLAT | Path/filename to a file containing WFC Lab flat | ClosedLoopWFC |
| WFC_SYS_FLAT | Path/filename to a file containing WFS system flats | ClosedLoopWFC |
| WFC_DM_SHAPE | Path/filename to a file containing DM shape | ClosedLoopWFC |
| BACK_N_AVG | RPG number of background images to average to create the backgrounds | SRT-RPG |
| FLAT_N_AVG | RPG number of flat images to average to create the backgrounds | SRT-RPG |

| | | |
|---|---|---|
| SRT_CONF_FILE | Path/filename to a file containing the SRT configuration, which contains key/value pairs:<br>- min/max loop gain<br>- default fudge gain for reconstructor<br>- number of PSD data points<br>- number of segments to average | SRT_RPG |
| SUBAPP_MASK_FILE | Sub-aperature mask file, fits format | SRT_RPG |
| NCPA_FILE | Non-common path file, if not using then this is a file full of 1's | LGS_PROCESSING |
| TEL_NUM_MODES | Number of telescope modes | Temporal Filtering and Combo, Telescope Offload |
| CENT_GAIN | X/Y Centroid gain (X, Y) | ClosedLoopWFC |
| TT_OFFSET | Tip/tilt offset, tip and tilt | ClosedLoopWFC |
| TT_MICS_PER_MAS | Microns per milliarcsec in X/Y | ClosedLoopWFC |
| DM_TT_COEFF | TT voltage calibration | ClosedLoopWFC |
| DM_COEFF | WFC voltage calibration, fits file format | ClosedLoopWFC |
| WFC APERTURE | WFC aperture pupil, fits file format | ClosedLoopWFC |
| TEL_LP_FILTER | Telescope offload low pass filter | Telescope Offload |
| TEL_OFF_RATE | Down sample rate for telescope offload | Telescope Offload |
| cb*.node | Each circular buffer is assigned to one memory node | Pipeline |
| cb*.capacity | All circular buffers store "capacity" historical frames of data before they are overwritten | Pipeline |
| hrt*Block.core | Most blocks will have a standard worker thread core assignment (specialized config in the case of multiple worker threads) | Pipeline |
| hrt*Block.cmdAddr | The IP address and port for all block command processors must be specified | Pipeline |
| hrt*Pipe.cmdAddr | The IP address and port for all hrtPipe command processors must be specified | Pipeline |
| hrt*BlockMap | Mapping of block processing tasks to computer hardware | Pipeline |

## 3.2 BLOCKS AND THEIR PIPES

Blocks are a way to group functionality together. It also allows for a fairly uniform way to customize HEART by include/excluding blocks. The Block names and descriptions follow.

Table 3-2 – Block Names and Descriptions

| Block Name | Description |
|---|---|

| | |
|---|---|
| **Standard WFS Input Block (hrtWfsInputBlock)** | This block handles inputs from WFS, etc. There is a part of that block that will be standard which interfaces with the rest of the HEART blocks |
| **High Order LGS Processing (hrtPixelProcBlock)** | Calibration of the pixels and conversion to gradients (if pixels are received) |
| **High Order Reconstruction (hrtHoReconBlock)** | Performs the MVM for the HO pixels |
| **Partial Vector Combination (hrtVecCombBlock)** | Only required if there are multiple High Order or High Order Truth Reconstruction blocks. They simply aggregate partial HO/HOT vectors from the Reconstruction blocks to produce a complete HO/HOT vector |
| **Low Order NGS Processing (hrtPixelProcBlock)** | Calibration of the pixels and conversion to gradients (if pixels are received) |
| **Low Order Reconstruction (hrtLoReconlock)** | Performs the MVM for the LO pixels |
| **Temporal Filtering & Combination (hrtTfcBlock)** | The main synchronization point in the HRT pipeline as it merges the HO, HOT, LO and LOT paths together |
| **Closed Loop Wavefront Correction (hrtClWfcBlock)** | Accepts DM error vector to command the DMs and TTSs |
| **Telescope Offload Block (hrtTelOffloadBlock)** | Performs the telescope offloading |
| **Telescope Mode Streaming (hrtStreamBlock)** | performs the streaming to the telescope |
| **Standard WFC Output Block (hrtWfsOutputBlock)** | This block handles outputs to the DM, TTS, etc. There is a part of that block that will be standard which interfaces with the rest of the HEART blocks |
| **Open Loop Wavefront Correction (hrtOlWfcBlock)** | Similar to the closed loop wavefront correction block but does not integrate the DM error vector |
| **Figure Processing (hrtFigProcBlock)** | Processes the figure WFS inputs |

A pipe can have 1 or more blocks in them. All pipes have the following functionality by default:

- Have a list of blocks associated with this pipe
- Will start a watchdog to make sure the pipes are working
- Have the ability to accept and send commands to all blocks in the pipe
    - START: will (re)start the blocks in reverse order and start the watchdog
    - sendCommand: Will send the command to all blocks in order
    - destroy: stops watchdog, then all blocks

There are a set of pipes used for various parts of the HEART RTC. These divisions and use of the pipeline can be modified. The default sets are defined in the following table.

Table 3-3 - HEART RTC Default Sets of Pipes

| Name | Description | Block Names |
|---|---|---|
| hrtHoPipe | All of the High Order LGS and NGS WFS processing, MVM, and output of HO vectors | Standard WFS Input Block (hrtWfsInputBlock), High Order LGS Processing |

| | | |
|---|---|---|
| | | (hrtPixelProcBlock), High Order Reconstruction (hrtHoReconBlock), Partial Vector Combination (hrtVecCombBlock) |
| hrtLoPipe | All of the Low Order NGS WFS processing, MVM, and output of LO vectors | Standard WFS Input Block (hrtWfsInputBlock), Low Order NGS Processing (hrtPixelProcBlock), Low Order Reconstruction (hrtLoReconlock), Partial Vector Combination (hrtVecCombBlock) |
| hrtTfcAndWfcPipe | Temporal Filtering and Combination of outputs from hrtHoPipe and hrtLoPipe, and generates Wavefront Corrector Open and Closed Loop Commands (for DM and TTS, and telescope offload) | Temporal Filtering & Combination (hrtTfcBlock), Closed Loop Wavefront Correction (hrtClWfcBlock), Telescope offload block (hrtTelOffloadBlock), Telescope Mode streaming (hrtStreamBlock), Standard WFC Output Block (hrtWfsOutputBlock) |
| hrtOpenWFCPipe | Open Loop Wavefront Corrector | Open Loop Wavefront Correction (hrtOlWfcBlock), Standard WFC Output Block (hrtWfsOutputBlock) |
| hrtFigurePipe | Figure WFS processing | Figure Processing (hrtFigProcBlock) |

## 3.3 CIRCULAR BUFFERS USED BY PIPES AND THEIR FORMATS

The following is description of the Circular Buffer formats. The formats include definitions from the configuration section, and a few additional formats that are then defined below the table. Where there are multiples of Blocks (N), the length of the circular buffer is increased by N.

Table 3-4 - Circular Buffer Formats

| Name | Description | Format |
|---|---|---|
| cbLgsRawPixels[1..N] | Raw LGS WFS pixels | [HO_WFS_SIZE] |
| cbLgsPixels[1..N] | Calibrated LGS WFS pixels | [NUM_HO_PIXELS] |
| cbLgsFlat[1..N] | LGS WFS flat fields | [HO_WFS_SIZE] |
| cbLgsDark[1..N] | LGS WFS darks | [HO_WFS_SIZE] |
| cbLgsBias[1..N] | LGS WFS bias | [HO_WFS_SIZE] |
| cbLgsSky[1..N] | LGS WFS sky background | [HO_WFS_SIZE] |
| cbLgsGradCoeff[X\|Y][1..N] | LGS WFS gradient coefficients | [2 * HO_WFS_SIZE] |

| | | |
|---|---|---|
| cbLgsGradThreshold[1..N] | LGS WFS Gradient Threshold | [HO_WFS_SA] |
| cbLgsGradWeight[1..N] | LGS WFS Gradient Weight Map | [HO_WFS_SIZE] |
| cbLgsGrad[X\|Y][1..N] | LGS WFS gradient | [HO_WFS_GRADS] |
| cbLgsSubApMask[1..N] | LGS WFS sub-aperture Mask | [HO_WFS_SA] |
| cbLgsSubApCoord[X\|Y][1..N] | LGS Sub-aperture X,Y coordinate map | [2 * HO_WFS_SA] |
| cbLgsSubApFlux[1..N] | LGS WFS sub-aperature flux | [HO_WFS_SA] |
| cbLgsFlux[1..N] | LGS WFS total flux | [1] |
| cbLgsTtErr | LGS TT err for the FSM | [2] |
| cbCmHo[1..N] | HO control matrices | [DM_SIZE] x [HO_WFS_GRADS] |
| cbImHo[1..N] | HO interaction matrices | [HO_WFS_GRADS] x [DM_SIZE] |
| cbPartialPol[1..N] | Partial HO POL gradient vectors | [HO_WFS_GRADS] |
| cbPartialHo[1..N] | Partial HO vectors | [DM_SIZE] |
| cbHo | Combined HO vector | [DM_SIZE] |
| cbHoLoopGain | High order loop gain | [1] |
| cbLoLoopGain | Low order loop gain | [1] |
| cbNgsRawPixels[1..N] | Raw NGS WFS pixels | [LO_WFS_SIZE] |
| cbNgsPixels[1..N] | Calibrated NGS WFS pixels | [LO_WFS_SIZE] |
| cbNgsFlat[1..N] | NGS WFS flat fields | [LO_WFS_SIZE] |
| cbNgsDark[1..N] | NGS WFS darks | [LO_WFS_SIZE] |
| cbNgsBias[1..N] | NGS WFS bias | [LO_WFS_SIZE] |
| cbNgsSky[1..N] | NGS WFS sky background | [LO_WFS_SIZE] |
| cbNgsGradCoeff[X\|Y][1..N] | NGS WFS gradient coefficients | [2 * LO_WFS_SIZE] |
| cbNgsGradThreshold[1..N] | NGS WFS Gradient Threshold | [1] |
| cbNgsGradWeight[1..N] | NGS WFS Gradient Weight Map | [LO_WFS_SIZE] |
| cbNgsGrad[X\|Y][1..N] | NGS WFS gradient | [LO_WFS_GRADS] |
| cbNgsFlux[1..N] | NGS WFS total flux | [1] |

| | | |
|---|---|---|
| cbCmLo | LO control matrices | [LO_NUM_MODES] x [LO_WFS_COUNT x LO_WFS_GRADS] x [2^ LO_WFS_COUNT - 1] |
| cbImLo | LO interaction matrix | [LO_WFS_COUNT x LO_WFS_GRADS] x [LO_NUM_MODES] |
| cbGradLo | Concatenated LO gradient vector | [LO_WFS_COUNT x LO_WFS_GRADS] |
| cbPOLLo | LO Pseudo Open Loop vector | [LO_WFS_COUNT x LO_WFS_GRADS] |
| cbLo | Combined LO output vector | [LO_NUM_MODES] |
| cbDmErr | Virtual DM error vector | [DM_SIZE] |
| cbWcShape | WC shape for POL feedback | [DM_SIZE] |
| cbClUnclipped | CL unclipped commands | [DM_SIZE] |
| cbClClipped | CL clipped commands | [DM_SIZE] |
| cbOlClipped | OL clipped commands | [DM_SIZE] |
| cbDm | DM commands | [DM_SIZE] |
| cbTt | TTS commands | [2] |
| cbFigRawPixelsX | Figure Raw pixels | [FIG_SIZE] |
| cbFigGrad | Figure Gradients | [FIG_GRAD] |
| cbFigDmVector | Figure DM Vector | [FIG_DM_SIZE] |
| cbCmTel | Telescope offload reconstructor | [TEL_NUM_MODES] |
| cbTelModes | Telescope offload modes | [TEL_NUM_MODES] |
| cbGlobalMem | Global memory | See Section 3.3.1 |
| cbAveHoIma | Averaged image in each LGS WFS sub-aperture | [HO_WFS_SIZE] |
| cbAveLoIma | Averaged image for each LO GS | [HO_WFS_SIZE] |
| cbAvLBWFSIma | Averaged image for each LO WFS sub-aperture | [LO_WFS_SIZE] |
| cbHoPSD | High-order residual PSD estimate | [srtNumPsdPts] |
| cbLoPSD | Low-order residual PSD estimate | [srtNumPsdPts] |
| cbHoGradPSD | High-order gradient PSD | [HO_WFS_GRADS] |
| cbLoGradPSD | Low-order gradient PSD | [LO_WFS_GRADS] |
| cbAveHoGrad[]* | For calibration purposes, when instructed (calibModeCmd) | [HO_WFS_GRADS] |

| | | |
|---|---|---|
| | the block will send the RPG averaged HO gradients | |
| cbAveLoGrad[]* | For calibration purposes, when instructed (calibModeCmd) the block will send the RPG averaged LO gradients | [LO_WFS_GRADS] |
| cbAveWcCmd[] | For calibration purposes, when instructed (calibModeCmd) the block will send the RPG averaged DM and TTS commands | [DM_SIZE] |
| cbWcOverride[] | For calibration purposes, when instructed (calibModeWc) the block will override the RTC control the DMs and TTS and apply the command as specified by the SRT-RPG. | [DM_SIZE] |

The following is a list of the circular buffers used in each of the pipes, and how they are used. This is mentioned here because the creation of the circular buffer is done outside of the block.

Table 3-5 - List of Circular Buffers for Each Pipe

| Pipe | Name | Usage | Description |
|---|---|---|---|
| **High Order Pipe (hrtHOPipe)** | cbLgsRawPixels[1..5] | Internal / intermediate | Raw LGS WFS pixels |
| | cbLgsPixels[1..5] | Internal / intermediate | Calibrated LGS WFS pixels |
| | cbLgsFlat[1..5] | Input | LGS WFS flafields |
| | cbLgsDark[1..5] | Input | LGS WFS darks |
| | cbLgsBias[1..5] | Input | LGS WFS bias |
| | cbLgsSky[1..5] | Input | LGS WFS sky background |
| | cbLgsGradCoeff[X\|Y][1..5] | Input | LGS WFS gradient coefficients |
| | cbCmHo[1..5] | Input | HO control matrices |
| | cbPartialPolHo[1..5] | Internal / output | Partial HO POL vectors |

| | | | |
|---|---|---|---|
| | cbPartialHo[1..5] | Internal / intermediate | Partial HO vectors |
| | cbImHo[1..N] | Internal / intermediate | HO interaction matrices |
| | cbLgsGradThreshold[1..N] | Internal / intermediate | LGS WFS Gradient Threshold |
| | cbLgsGradWeight[1..N] | Internal / intermediate | LGS WFS Gradient Weight Map |
| | cbLgsSubApFlux[1..N] | Internal / intermediate | LGS WFS sub-aperature flux |
| | cbLgsFlux[1..N] | Internal / intermediate | LGS WFS total flux |
| | cbHoLoopGain | Internal / intermediate | High order loop gain |
| | cbHo | Output | Combined HO vector |
| | cbAveHoIma | Output | HO statistics |
| | cbHoPSD | Output | Ave HO PSD |
| | cbHoGradPSD | Output | HO PSD for each Grad |
| | cbAveHoGrad | Output | Ave HO Grad PSD |
| **hrtLoPipe** | cbNgsRawPixels[1..N] | Internal / intermediate | Raw NGS WFS pixels |
| | cbNgsPixels[1..N] | Internal / intermediate | Calibrated NGS WFS pixels |
| | cbNgsFlat[1..N] | Input | NGS WFS flat fields |
| | cbNgsDark[1..N] | Input | NGS WFS darks |
| | cbNgsBias[1..N] | Input | NGS WFS bias |
| | cbNgsSky[1..N] | Input | NGS WFS sky background |
| | cbNgsGradCoeff[X\|Y][1..N] | Input | NGS WFS gradient coefficients |
| | cbCmLo | Input | LO control matrix |
| | cbGradLo | Internal / intermediate | Concatenated LO gradient vector |
| | cbPOLLo | Internal / intermediate | LO Pseudo Open Loop vector |
| | cbImLo[1..N] | Internal / intermediate | LO interaction matrices |
| | cbNgsGradThreshold[1..N] | Internal / intermediate | NGS WFS Gradient Threshold |
| | cbNgsGradWeight[1..N] | Internal / intermediate | NGS WFS Gradient Weight Map |
| | cbNgsGrad[X\|Y][1..N] | Internal / intermediate | NGS WFS gradient |
| | cbNgsFlux[1..N] | Internal / intermediate | NGS WFS total flux |
| | cbLoLoopGain | Internal / intermediate | Low order loop gain |
| | cbLo | Output | Combined LO vector |
| | cbAveLoIma | Output | Ave image LO GS |
| | cbAvLBWFSIma | Output | Ave LBWFS sub-app |
| | cbLoPSD | Output | Ave LO PSD |
| | cbLoGradPSD | Output | LO PSD for each Grad |

| | | | |
|---|---|---|---|
| | cbSrtRpgSettings | Input | SRT-RPG Settings |
| **hrtTfcAndWfcPipe** | cbDmErr | Internal / intermediate | Virtual DM error vector |
| | cbWcShape | Output | WC shape for POL feedback |
| | cbClUnclipped | Output | CL unclipped commands |
| | cbClClipped | Output | CL clipped commands |
| | cbCmTel | Input | Telescope offload reconstructor |
| | cbTelModes | Output | Telescope offload modes |
| | cbFigDmVector | Input | DM Figure DM Vector |
| | cbDm | Output | DM commands |
| | cbTt | Output | TTS commands |

### 3.3.1 Global Memory

The Global Memory circular buffer is unique in that there will only be 1 of them. Each portion of the block is written to by 1 block, but read by multiple blocks.

All blocks have a “state” variable which defines their current state.

Also, the notable and alarm status are included.

Table 3-6 - Global Memory Circular Buffer Contents

| Section | Name | Description | Type |
|---|---|---|---|
| **generalConfig (General information from the configuration file)** | hoWfsCount | # HO WFS | int |
| | hoWfsSize | Max number of subapertures in each HO WFS | [int][int] |
| | hoWfsGradCount | Number of slopes | int |
| | hoWfsChunks | Size of HO WFS pixel data chunks | int |
| | loWfsCount | # LO WFS | int |
| | loWfsSize | Max number of subapertures in each LO WFS | [int][int] |
| | loWfsGradCount | Number of slopes | int |
| | loWfsChunks | Size of LO WFS pixel data chunks | int |
| | loNumModes | Number of modes | int |
| | dmCount | Number of DMs | int |
| | dmSize | Size of each DM | [int][int] |
| | flatFile | Flat path/filename | string |
| | darkFile | Dark path/filename | string |
| | biasFile | Bias path/filename | string |
| | refOffsFile | Reference offsets | string |
| | hardwareMapFile | Hardware mapping path/filename | string |
| | wfcFlatLabFile | Wavefront correct flat and lab filenames | string |
| | telNumModes | Telescope number of modes | int |
| | srtMinLoopGain | Minimum loop gain | |
| | srtMaxLoopGain | Maximum loop gain | |
| | srtReconFudge | Default fudge gain for each reconstructor | |
| | srtNumPsdPts | Number PSD data points | |
| | srtSegToAve | Number segments to average for PSDs | |
| | lgshrtHoPipeIP | LGS High Order pipe IP Addresses | string[], max 6 |
| | ngshrtHoPipeIP | NGS High Order pipe IP Addresses | string[], max 3 |
| | ngshrtLoPipeIP | NGS Low Order pipe IP Addresses | string[], max 6 |

| | | | |
|---|---|---|---|
| | hrtTfcAndWfcPipeIP | Temporal Filtering & Combination and WC IP Addresses | string[], max 6 |
| | hrtOtherPipeIP | This is a custom pipe, eg. Slow Focus pipe, IP Addresses | string[], max 6 |
| **srtRpg** | state | Current state of the block | |
| | msgString | Message string, used for debugging and reporting any errors | |
| | controlMatrixFrameNum | Control Matrix Frame Number, active cm, -99 for none | |
| **commandHandler** | enableStatistics | Statistics gathering is enabled and ready to receive | |
| | cmd | RTC command state | enum: (UNINITIALIZED, READY, BUSY) |
| | mode | Selected mode, wide or narrow field | enum |
| | rates | Rates of all WFS | [int][int] |
| | loopState | State of loops | int |
| | overallState | Overall state of the RTC | |
| | rpgDate | Flag indicating if the RTC as all the required data from the RPG to start the pipeline | boolean |
| | lgsBackground | Flag indicating if the LGS background task is currently active | boolean |
| | pol | Indicates whether pseudo open-loop (POL) feedback is enabled. When POL is disabled then the uncontrolled mode cleanup path is enabled. Nominally POL will be used in LGS mode and disabled for NGS mode | boolean |
| | unsavedConfig | Flag indicating if any configuration parameters have been changed, via the configParamSet or calibLgsBackground command, but has not yet been saved | boolean |
| | rts-mode | Real-Time Streamer Mode | |

| | | | |
|---|---|---|---|
| | optim_slope | Flag indicates if slope optimization is enabled | boolean |
| | optim_thres | Flag indicates if threshold optimization is enabled | boolean |
| | optim_weight | Flag indicates if weight optimization is enabled | boolean |
| | optim_frameRate | Flag indicates if frameRate optimization is enabled | boolean |
| | optim_loopGain | Flag indicates if loopGain optimization is enabled | boolean |
| | event_ lowFlux | Indicates insufficient flux detected on a detector | |
| | event_ log | Includes the output of the command handler, and it based on the current debut level | |
| | event_ overallPipelineStatus | A combination of all of the block/pipeline status | |
| | event_ pipelineDelay | Indicates part or all of the RTC pipeline is running slow | |
| | alarm_ watchdog | Indicates some part of the RTC has become unresponsive | |
| | alarm_ pipelineDelay | Indicates part of all of the RTC pipelines is running slow (exception is when pipeline rates are changed and it is settling) | |
| | alarm_ pipelineFault | Indicates part or all of the RTC pipeline is not running | |
| **ngsProcessing** | blockNum | Block number (used where there are multiple blocks of this type) | |
| | state | Current state of the block | enum |
| | msgString | Message string, used for debugging and reporting any errors | string |
| | loopParms | Possible loop parameters<br>wc_int - wavefront corrector integrator (= 0)<br>wc_tt_lpf - TTS filter states (= 0)<br>wc_tel_lpf - Telescope offload | |

| | | | |
|---|---|---|---|
| | | high-pass filter states (= 0)<br>ho_psd - High-order PSD & WFE (discard gathered statistics)<br>ho_turb - other turbulence parameters (discard gathered statistics)<br>lo_mode_hpf - low-order (Tier 1 & 2) high-pass filter states (= 0)<br>lo_mode_lpf - low-order truth (Tier 3 & 3F) low-pass filter states (= 0)<br>lo_ssm_lpf - NGS SSM offload low-pass filter state (= 0)<br>lo_psd - Low-order PSD & WFE (discard gathered statistics)<br>ngs_opt_gain - NGS optical gain (ngsState.opticalGain = 1)<br>lgs_focus_lpf - LGS focus low-pass filter states (= 0)<br>lgs_opt_i0 - LGS i0 (discard gathered statistics, i.e. time-series of ib frames)<br>lgs_tt_int - LGSF FSM integrator (= 0)<br>lgs_tt_psd - LGS TT PSD (discard gathered statistics) | |
| | setNgsCentroidingErr | Enable/disable NGS centroiding error corrections | |
| | calibrating | Indicates if currently performing calibration | |
| | darkFilename | Path/Filename used for dark file | |
| | enableStats | Enabled/disabled the gathering of statistics: ave image for each image, LO GS | |
| | flatFilename | Path/Filename used for dark file | |
| | event_Failed | Indicates error while working | |

| | | | |
|---|---|---|---|
| | event_ ngsPixelMissing | At least one NGS pixel UDP datagram was missing from the frame | |
| | event_ ngsOutOfOrder | At least one NGS pixel UDP datagram was received out of order | |
| | event_ ngsBadCrc | A bad CRC was detected in at least one NGS pixel UDP datagram | |
| | event_ ngsLowFlux | Insufficient flux detected on NGS | |
| | event_ ngsOiwfsPixelMissing | At least one OIWFS pixel UDP datagram was missing from the frame | |
| | event_FailedTrigger | Indicates a failed trigger semaphore | |
| **lgsProcessing(an array of structs, for each number of blocks enabled)** | blockNum | Block number (used where there are multiple blocks of this type) | |
| | state | Current state of the block | enum |
| | msgString | Message string, used for debugging and reporting any errors | string |
| | setLgsTt | Enable/disable the sending of LGS TT (or Centering errors) to the Laser Fast Steering Mirrors (Jitter Mirror) | |
| | calibrating | Indicates if currently performing calibration | |
| | backFilenameInUse | Path/Filename used for dark file | |
| | flatFilenameInUse | Path/Filename used for dark file | |
| | enableStats | Enabled/disabled the gathering of statistics: ave image for each sub-ap, ave image for each LO GS, average image | |
| | event_ lgsTtCorrectionLost | Indicates lost LGS TT correction | |
| | event_ lgsFocusConvergeError | Indicates missing some or all LGS focus errors | |
| | event_Failed | Indicates error while working | |
| | event_FailedTrigger | Indicates a failed trigger semaphore | |
| **hoRecon (an array of structs, for each number of blocks enabled)** | blockNum | Block number (used where there are multiple blocks of this type) | |

| | | | |
|---|---|---|---|
| | state | Current state of the block | enum |
| | msgString | Message string, used for debugging and reporting any errors | string |
| | enableLgsHoWfs | Enable/disable using LGS WFS for HO calculations | |
| | event_Failed | Indicates error while working | |
| | event_ hoCorrectionLost | Indicates HO error vector cannot be computed | |
| | event_ hoErrVectLost | Indicate HO error vector is late | |
| | event_ loErrVectLost | Indicate LO error vector is late | |
| | event_FailedTrigger | Indicates a failed trigger semaphore | |
| **loRecon (an array of structs, for each number of blocks enabled)** | blockNum | Block number (used where there are multiple blocks of this type) | |
| | state | Current state of the block | |
| | msgString | Message string, used for debugging and reporting any errors | |
| | enablePOLFeedback | POL Feedback is enabled | |
| | enableMVMOutput | MVM output is enabled | |
| | enableMVMGainOptim | When set perform gain optimization of the MVM | |
| | mvmGainOptim | optimization Gain | |
| | numValidWfsCnt | Number of "no valid" WFS gradients received in a row | |
| | loWfsMask | Valid WFS as given by the input block | |
| | event_NoValidWfs | Indicates for 1 iteration no valid WFSs | |
| | event_Failed | Indicates error while working | |
| | event_ lotLost | Indicates LOT modes cannot be computed | |
| | event_loDegrade | Indicates only some of the LO modes can be computed | |
| | event_ loNgsLost | Indicates LO modes, based on the Tier 0 NGS modes, cannot be computed | |
| | event_FailedTrigger | Indicates a failed trigger semaphore | |

| | | | |
|---|---|---|---|
| **tempFilterCombo (Temporal Filter & Combo)** | state | Current state of the block | enum |
| | msgString | Message string, used for debugging and reporting any errors | string |
| | loPipelineIntegrators | LO Pipeline control (start, stop, resume, reset) only the control lop | |
| | hoPipelineIntegrator | HO Pipeline control (start/stop/resume/reset) | |
| | enableNgsFieldDist | Enable/disable the sending of field distortion modes to the high order loop | |
| | enableNgsFieldRotOffset | Enable/disable NGS Field Rotation corrections to the System Controller | |
| | instRotationErr | Instrument rotator error | |
| | sync | Indicates sync status | |
| | event_Failed | Indicates error while working | |
| | event_ loCorrectionLost | Indicates LO error vector cannot be computed | |
| | event_ lgsPolGradLost | Indicates missing some or all LGS POL gradients | |
| | event_ lgsDmShapeLost | Indicates lost LGS DM shape feedback | |
| | event_ hoDmVectOutOfOrder | At least one HO DM vector UDP datagram was received out of order | |
| | event_ hoDmVectBadCrc | A bad CRC was detected in at least one HO DM vector UDP datagram | |
| | event_FailedTrigger | Indicates a failed trigger semaphore | |
| **telOffload (Telescope offload)** | state | Current state of the block | enum |
| | msgString | Message string, used for debugging and reporting any errors | string |
| | comma | Comma correction for M2 | |
| | focus | Focus correction for M2 | |
| | enableComaFocus | Enable/disable the sending of coma and focus corrections to the System Controller for | |

| | | | |
|---|---|---|---|
| | | the Secondary Control System. | |
| | enablePrimaryFigure | Enable/disable the sending of primary mirror figure corrections to the RTC System Controller | |
| | enableAveTTtoSCS | Enable/disable the average tip tilt corrections to the Secondary Control Mirror | |
| | telOffloadTt | Telescope TT offloading | |
| | telOffloadFocus | Telescope focus offloading | |
| | telOffloadMag | Telescope maginification offloading | |
| | telOffloadMode | Telescope M1 modes offloading | |
| | event_Failed | Indicates error while working | |
| | event_FailedTrigger | Indicates a failed trigger semaphore | |
| **clWfc (closed loop wavefront corrector)** | state | Current state of the block | enum |
| | msgString | Message string, used for debugging and reporting any errors | string |
| | stateLoIntegrators | Enable/disable/reset LO Integrator (only Lo or Ho – they are the same) | |
| | stateHoIntegrators | Enable/disable/reset HO Integrator (only Lo or Ho – they are the same) | |
| | hoIntVal | Integration value | |
| | enableTipTiltOutput | Enable/disable TTS output | |
| | enableClwfc | Enable output to DMs | |
| | event_Failed | Indicates error while working | |
| | event_ ttsSenseLost | Indicates failure to get TTS postion | |
| | event_dm0Clip, event_dm1Clip, event_dm2Clip | Indicates DM was excessively clipped | |
| | event_ lgsTtConvergeError | Indicates LGS TT error has not converged in the expected amount of time | |
| | event_FailedTrigger | Indicates a failed trigger semaphore | |
| **standardWfsInputBlock** | state | Current state of the block | enum |

| | | | |
|---|---|---|---|
| | msgString | Message string, used for debugging and reporting any errors | string |
| | type | Type of Input, eg, WFS | enum |
| | id | Designator, eg: 1, 2, 3, 4 | int |
| | hardware | Hardware attached | |
| | event_Failed | Indicates error while working | |
| | event_FailedTrigger | Indicates a failed trigger semaphore | |
| **telescopeStream** | state | Current state of the block | enum |
| | msgString | Message string, used for debugging and reporting any errors | string |
| | enableStream | Enabled output stream | |
| | location | Where to send the data | |
| | type | Type of data sending | |
| | event_Failed | Indicates error while working | |
| | event_FailedTrigger | Indicates a failed trigger semaphore | |
| **standardOutputBlock** | state | Current state of the block | enum |
| | msgString | Message string, used for debugging and reporting any errors | string |
| | location | Where to send the data | |
| | type | Type of data sending | |
| **olWfc (openLoop Wavefront Corrector)** | state | Current state of the block | enum |
| | msgString | Message string, used for debugging and reporting any errors | string |
| | event_Failed | Indicates error while working | |
| | event_FailedTrigger | Indicates a failed trigger semaphore | |
| **Figure Processing** | state | Current state of the block | enum |
| | msgString | Message string, used for debugging and reporting any errors | string |
| | event_Failed | Indicates error while working | |
| | event_FailedTrigger | Indicates a failed trigger semaphore | |
| **partialVector** | TBD | | |

## 3.4 COMMANDS TO BLOCKS

The commands to the blocks are listed in the table below.

Table 3-7 - List of Commands to Blocks

| Pipe | Blocks | Description | Commands | CB |
|---|---|---|---|---|
| **High Order Pipe (hrtHOPipe)** | Standard WFS Input Block (**hrtWfsInputBlock**) (xN) | Receive input from 1 WFS and put into CB | RCV_DATA (Start receiving data) | cbLgsRawPixelsX (X=1..N, one for each HO WFS) |
| | High Order NGS Processing (**hrtPixelProcBlock**) (xN) | Continuously monitor CB for input streams, calibrate (bias/dark/flat) and write each subapp to CB and trigger | PIXEL_PROC_UPDATE_CAL (Search for and update as needed new information in cbLgsFlat, cbLgsDark, cbLgsBias, cbLgsSky, cbLgsGradCoeff) | cbLgsPixels cbLgsFlat, cbLgsDark, cbLgsBias, cbLgsSky, cbLgsGradCoef cbLgsTtErr |
| | High Order LGS Processing (**hrtPixelProcBlock**) (xN) | Continuously monitor CB for input streams, calibrate and write each subapp to CB and trigger | PIXEL_PROC_UPDATE_CAL (Search for and update as needed new information in cbLgsFlat, cbLgsDark, cbLgsBias, cbLgsSky, cbLgsGradCoeff) ENABLE_LGS_FSM (enable LGS offload) LGS_GRAD_OPTIM (LGS gradient optimization) | cbLgsPixels cbLgsFlat, cbLgsDark, cbLgsBias, cbLgsSky, cbLgsGradCoef cbLgsTtErr |
| | High Order Reconstruction (**hrtHoReconBlock**) (xN) | Perform streamed MVM and write to CB & project WC to gradient space | MVM_WC_CB (Provide reference to WF shape CB created externally) | cbCmLo |
| | | | MVM_UPDATE_MATRIX( Start using new control matrix in cbCmHo) | cbCmLo |
| | | | MVM_USE_POL (Activate/Deactivate POL feedback) | cbPartialPolHo, cbPartialHo |
| | High order truth reconstruction (**hrtHoReconBlock**) | Same as HO Reconstruction | MVM_WC_CB (Provide reference to WF shape CB created externally) | cbLgsGradX, cbPartialHoX, cbWcShape (from) if using POL: cbCmHoX |
| | | | MVM_UPDATE_MATRIX (Start using new control matrix in cbCmHo) | cbCmLo |
| | Partial Vector Combination (**hrtVecCombBlock**) | (Partial Vector Combination) that produces a single HO | | |

| | | | | |
|---|---|---|---|---|
| | | vector from the five partial vectors. | | |
| **Low Order Pipe (hrtLoPipe)** | Standard WFS Input Block (**hrtWfsInputBlock**) (xN) | Receive input from 1 WFS and put into CB | | cbNgsRawPixelsX (X=1..N, one for each LO WFS) |
| | Low Order NGS Processing (**hrtPixelProcBlock**) (xN) | Continuously monitor CB for input streams, calibrate (bias/dark/flat) and write each subapp to CB and trigger | PIXEL_PROC_UPDATE_CAL (Search for and update as needed new information in cbNgsFlat, cbNgsDark, cbNgsBias, cbNgsSky, cbNgsGradCoeff) | hrtPixelProcBlock[1..N] |
| | Low order reconstruction (**hrtLoReconBlock**) (x1) | (LO reconstruction) that produces LO vectors from the gradients produced by the preceding hrtVecCombBlock using matrix vector multiplication. | MVM_UPDATE_MATRIX (Start using new control matrix in cbCmLo) | cbCmLo |
| | | | MVM_USE_POL (Activate/Deactivate POL feedback) | cbImLo<br>cbPOLLo |
| | Partial Vector Combination (**hrtVecCombBlock**) (x1) | (Partial Vector Combination) that produces a single LO gradient vector from the individual gradient vectors. | | cbWcShape<br>and dependent on where this is used |
| **Temporal Filtering & Combination and Wavefront Correction (hrtTfcAndWfcPipe)** | Temporal Filtering & Combination (**hrtTfcBlock)** (x1) | Merges the LO and HO vectors, includes WFC feedback, and produces the DM Error vector | lgsServoLoopLag (Servo lag for the LGS loop) | cbLo<br>cbHo<br>cbDmErr<br>cbWcShape |
| | | | filterHo (Parameters for filter applied to LO vector) | hrtTfcBlock |
| | Closed Loop Wavefront Correction (**hrtClWfcBlock**) (x1) | Derives DM and TTS commands from the DM Error Vector, and also generates WFC shape vector feedback for the hrtTfcBlock and also the hrtHoPipe and hrtLoPipe. | | cbDmErr<br>cbWcShape<br>cbClClipped<br>cbFigDmVector<br>cbDm<br>cbTt |
| | Telescope offload block (**hrtTelOffloadBlock)** (x1) | Projects unclipped DM commands into telescope modes which are then passed through a temporal filter. The filtered tip and tilt modes are sent directly to the secondary control system at a reduced rate | MVM_UPDATE_MATRIX (Start using new control matrix in cbCmTel) | cbPartialPolHo<br>cbClClipped<br>cbClUnclipped<br>cbCmTel<br>cbTelModes |
| | Telescope Mode streaming (**hrtStreamBlock)** (x1) | Down samples the remaining filtered low order modes and sends them to the Command | enableComafocusErr (Enable/disable comma and focus offload) | cbCmTel<br>cbTelModes |

| | | | | |
|---|---|---|---|---|
| | | Handler where they are forwarded on to the System Controller. | | |
| | Standard WFC Output Block (**hrtWfsOutputBlock**) (xN) | Output data to a WFC | | cbDm<br>cbTt |
| Open Loop (**hrtLoWFCPipe**) | Open Loop Wavefront Correction (hrtOlWfcBlock) (x1) | User DM Error vector to get unclipped OL command and control DM and TTS. | | cbDmErr<br>cbOlClipped |
| | Standard WFC Output Block (**hrtWfsOutputBlock**) (xN) | Output data to a WFC | | cbDm |
| Figure WFS (**hrtFigurePipe**) | Standard WFS Input Block (**hrtWfsInputBlock**) (xN) | Receive input from 1 WFS and put into CB | | cbFigRawPixelsX<br>cbFigGrad |
| | Figure Processing (**hrtFigProcBlock**) | Process the figure wavefront sensor pixels | | cbFigDmVector |

Additional commands from the Command Handler to the Blocks are below.

Table 3-8 - Additional Commands from Command Handler to Blocks

| Block | Command | Description |
|---|---|---|
| **All** | config | Receive configuration parameter read form assembly configuration file |
| **LGS Processing** | configQuery | Report the current value of all pipeline configuration parameters |
| **LGS Processing,NGS Processing** | mode | Change AO mode of the pipeline |
| **All** | simulate | Change simulation state of external system connections |
| **LGS Processing** | calibLgsBackground | Rake new LGS WFS background images |
| **LGS Processing** | calibModePixel | Stream pixels to RPG for calibration |
| **LGS Processing** | calibModeGrad | Stream gradients to RPG for calibration |
| **LGS Processing** | calibModeCmd | Stream wavefront corrector commands to RPG for calibration |
| **LGS Processing** | calibModeWc | Enable/disable wavefront corrector command stream from the RPG for calibration |
| **All** | pipelineCntrl | Start/stop the RTC pipeline |
| **SFS Reconstructor** | loopLgsFocus | Enable/disable LGS focus loop (offloading to System Controller) |
| **LGS Processing** | loopLgsTt | Enable/disable LGS TT loop (commands to the LGSF) |
| **HO Reconstructor** | loopHigh | Enable/disable HO loop |
| **LO Reconstructor** | loopLoOrder | Enable/disable LoOrder loop |
| **Telescope Offload** | offloadTcs | Enable/disable TCS offloading |

| | | |
|---|---|---|
| **Command Handler** | loopParamReset | Reset loop parameter or statistic |
| **LGS Processing** | lgsfFsmZero | Zero the LGSF FSMs |
| **Closed Loop Corrector** | dmFlatten | Flatten the DMs |
| **Closed Loop Corrector** | ttsZero | Zero the TTS |
| **Command Handler** | test | Runs the automatic diagnostic self-test sequence |
| **All** | bufferDump | Dump circular buffer to RTS |

## 3.5 BLOCK TO SRT RPG STATISTICS GATHERING AND INFORMATION

The following tables describes the RPG data that is gathered, and the block that it is gathered in. And if appropriate, the rate is included (in some cases the rate is indicated in the configuration file). (These circular buffers are included in the master list in Section 3.3.

Table 3-9 - Block to SRT RPG (statistics)

| Circular Buffer | Description | HEART doc section | Rate (nominal) | Block |
|---|---|---|---|---|
| cbAveHoIma | Averaged image in each LGS WFS sub-aperture | Gradient Optimization (4.2.1) & 3.1.1 | 0.2 Hz or slower | High Order LGS Processing (**hrtPixelProcBlock**) |
| cbAveLoIma | Averaged image for each LO GS | Gradient Optimization (4.2.1) | 0.2 Hz or slower | Low Order LGS Processing (**hrtPixelProcBlock**) |
| cbHoPSD | High-order residual PSD estimate<br>• Time and space-averaged PSD of residual WFE (in DM actuator space)<br>• Time-averaged PSD estimates are computed for a subset of DM actuators<br>• All the time-averaged PSDs are averaged before being sent to RPG (so one PSD estimate is sent) | HO loop gain optimization (4.2.2.1) | 0.2 Hz or slower | Temporal Filtering & Combination (**hrtTfcBlock**) |

| | | | | |
|---|---|---|---|---|
| cbLoPSD | Low-order residual PSD estimate<br>• Time-averaged PSD of residual modal coefficients<br>• One time-averaged PSD estimate is computed for each LO mode and sent to RPG | LO loop gain optimization (4.2.2.2) | 0.2 Hz or slower | Temporal Filtering & Combination (**hrtTfcBlock**) |
| cbHoGradPSD | High-order gradient PSD estimate<br>• Time-averaged PSD for each HO gradient (two gradients per sub-aperture) | Tomographic Reconstructor *R* (4.3.1.1.1) | 0.2 Hz or slower | High Order NGS Processing (**hrtPixelProcBlock**) |
| cbLoGradPSD | Low-order gradient PSD estimate Time-averaged PSD for each NGS (two gradients per NGS) | Least Square (LS) Reconstructor (4.3.1.2) | 0.2 Hz or slower | Low Order NGS Processing (**hrtPixelProcBlock**) |
| cbAveHoGrad[]* | For calibration purposes, when instructed (calibModeCmd) the block will send the RPG averaged HO gradients | 4.4.2 Interaction matrix calibration | | High Order NGS Processing (**hrtPixelProcBlock**) |
| cbAveLoGrad[]* | For calibration purposes, when instructed (calibModeCmd) the block will send the RPG averaged LO gradients | 4.4.2 Interaction matrix calibration | | Low Order NGS Processing (**hrtPixelProcBlock)** |
| cbAveWcCmd[] | For calibration purposes, when instructed (calibModeCmd) the block will send the RPG averaged DM and TTS commands | 4.4.2 Interaction matrix calibration | | Closed Loop Wavefront Correction (**hrtClWfcBlock**) |

*For calibration purposes

## 3.6 SRT-RPG TO HRT BLOCK INFORMATION

The following table indicates the information received from the SRTC (via the RPG Handler) and the block that uses the information.

Table 3-10 - SRT to HRT Block

| SRT Data | Block |
|---|---|

| | |
|---|---|
| **Control matrices** | High Order Reconstruction<br>And Low Order Reconstruction |
| **Interaction Matrix** | High Order Reconstruction<br>And Low Order Reconstruction |
| **Sub-Aperture Mask** | High Order NGS Processing |
| **Gradient Threshold** | High Order NGS Processing<br>And Low Order NGS Processing |
| **Gradient Weight Map** | High Order NGS Processing<br>And Low Order NGS Processing |

## 3.7 ENABLE/DISABLE FLAGS AT A BLOCK LEVEL

Each block has individual flags that can enable or disable functionality within that block. They are detailed in the following table and reported by those blocks. Please note that these are also published in the Global Memory (Section 3.3.1).

Table 3-11 - Block Enable/Disable Flags

| Description | Block |
|---|---|
| Enable/disable NGS centroiding error corrections | Low Order NGS Processing |
| Enable/disable NGS Field Rotation corrections to the System Controller | Temporal Filtering & Combination |
| Enable/disable TTS output | Closed Loop Wavefront Correction |
| Enable/disable the average tip tilt corrections to the Secondary Control Mirror | Telescope offload block |
| Enable/disable the sending of field distortion modes to the high order loop | Temporal Filtering & Combination |
| Enable/disable the sending of LGS Focus errors to the System Controller | SFS Recon |
| Enable/disable using LGS WFS for HO calculations | Closed Loop Wavefront Correction |
| Enable/disable the sending of LGS Centering (or LGS TT) errors to the Laser Fast Steering Mirrors | Telescope offload |
| Enable/disabled the sending of Deformable Mirror shapes to Deformable Mirrors | Closed Loop Wavefront Correction |
| Enable/disable the sending of coma and focus corrections to the System Controller for the Secondary Control System. | Telescope offload |
| Enable/disable the sending of primary mirror figure corrections to the RTC System Controller | Telescope offload |
| Enable/disable/reset LO Integrator | LO Reconstructor |
| Enable/disable/reset HO Integrator | HO Reconstructor |
| LO Pipeline control (start, stop, resume, reset) | LO Reconstructor |
| HO Pipeline control (start/stop/resume/reset) | HO Reconstructor |