|  |  |
| --- | --- |
| moons_logo.jpg | |
| Document Title | Process Functional Specification |
| Document Number |  |
| Issue | 0.2 |
| Date |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Prepared By | S Watson | Signature |  |
|  | Date |  |
| Approved By |  | Signature |  |
|  | Date |  |
| Released By |  | Signature |  |
|  | Date |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Contributors | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Change Record**

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Date** | **Sections Affected** | **Change Description** |
| 0.1 | 15-02-2019 | All | First draft |
| 0.2 | 26-02-2018 |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

[1 INTRODUCTION 1](#_Toc2089559)

[2 PREREQUISITES 2](#_Toc2089560)

[3 Verification process 3](#_Toc2089561)

[3.1 General rules 3](#_Toc2089562)

[3.2 Initialisation 3](#_Toc2089563)

[3.3 Verification rig setup 3](#_Toc2089564)

[3.4 Collision detection 4](#_Toc2089565)

[3.5 Limit characterisation 5](#_Toc2089566)

[3.6 Pupil alignment 5](#_Toc2089567)

[3.7 Metrology calibration 6](#_Toc2089568)

[3.8 Metrology height 6](#_Toc2089569)

[3.9 Datum repeatability 7](#_Toc2089570)

[3.10 Positional repeatability and gearbox calibration 7](#_Toc2089571)

[3.11 Positional verification 8](#_Toc2089572)

[4 Hardware control specification 9](#_Toc2089573)

[4.1 FPU control 9](#_Toc2089574)

[4.2 Rotary stage 9](#_Toc2089575)

[4.3 Linear stage 10](#_Toc2089576)

[4.4 Lamps and control DAQ 10](#_Toc2089577)

[4.5 Camera control 10](#_Toc2089578)

[5 Image analysis specification 11](#_Toc2089579)

[5.1 Find metrology targets from posrep camera 11](#_Toc2089580)

[5.2 Find metrology targets from metcal camera 11](#_Toc2089581)

[5.3 Find backlit fibre from metcal camera 12](#_Toc2089582)

[5.4 Find metrology target heights from metht camera 12](#_Toc2089583)

[5.5 Find projected spot from pupaln camera 13](#_Toc2089584)

[6 Performance parameter calculations 14](#_Toc2089585)

[6.1 Metrology target calibration 14](#_Toc2089586)

[6.2 Pupil alignment 14](#_Toc2089587)

[6.3 Datum repeatability 14](#_Toc2089588)

[6.4 Positional repeatability 14](#_Toc2089589)

[6.5 Gearbox correction calibration 15](#_Toc2089590)

[6.6 Positional accuracy 15](#_Toc2089591)

[7 Database 16](#_Toc2089592)

**LIST OF FIGURES**

|  |  |
| --- | --- |
| FIGURE | PAGE |

[Figure 1: FPU Verification Process Flowchart 1](#_Toc2089593)

**LIST OF TABLES**

|  |  |
| --- | --- |
| TABLE | PAGE |

**No table of figures entries found.**

# 

# INTRODUCTION

This document analyses the “FPU Verification Process Flow Chart”, below, and breaks it into smaller tasks, which are described in detail.

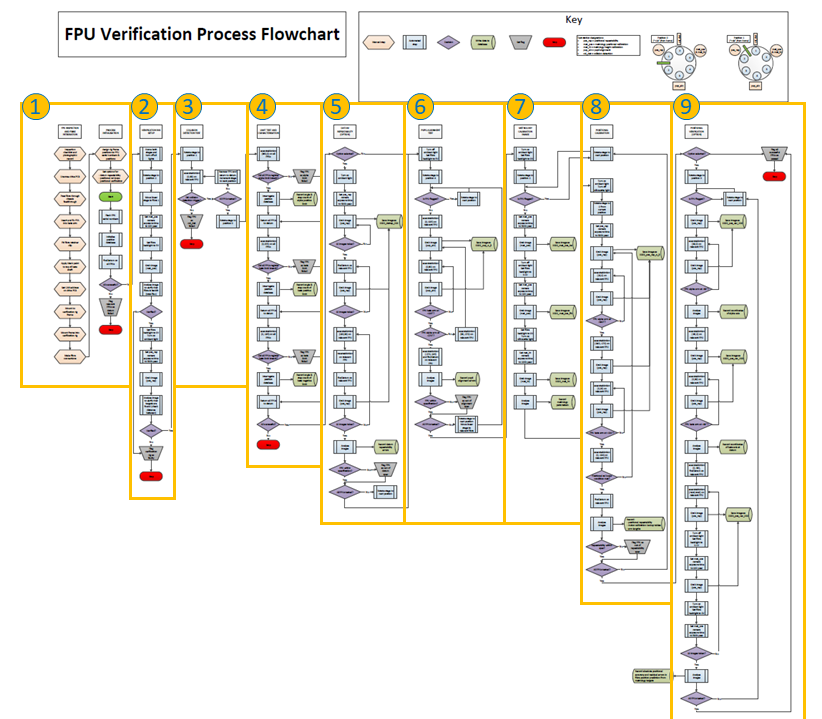


Figure 1: FPU Verification Process Flowchart

# PREREQUISITES

The following things need to be completed before the tests can be carried out.

* Software to control each camera, take exposures and save the images to a file.
* Metrology camera distortion calibrated using a rectangular pattern of dots. This process generates a set of distortion coefficients which can be used to rectify each camera image.
  + Primary plan: calibration the cameras with Holger’s software
  + Backup plan: calibrate the cameras with openCV library functions.
* Software to control the Thorlabs turntable.
* Software to control the linear stage.
* Software to control the lamps.
* FPU control parameters (minimum and maximum speed and acceleration) agreed.
* The turntable, linear stage, lights and cameras have all been tested and verified to work.  
  There should a self-test procedure which verifies all the components are responding.

# Verification process

## General rules

* When moving the rotary stage, the last successful operation from all FPUs must have been a findDatum. The only exception to this is prior to running the Collision Detection Test.
* Images taken should not be deleted or overwritten.
* Where red dots are indicated ● the intention is that the test will stop with an error and the operator will take immediate action. In all other cases, unless otherwise stated, it should be assumed that errors are handled by the software and the process should continue.
* Where a process is manual (i.e. not controlled by software), it is stated in *italics.*

## Initialisation

Purpose – this is the process where a user loads FPUs into the verification rig, flashes the electronics with the correct serial number and ensures that the FPU is responding to commands by running a findDatum operation.

*For each fibre positioner received*

*Note the unique serial number assigned to that fibre positioner (etched onto chassis)*

*If FPU position is not at or very close to datum then*

*Manually initialise FPU to the point where it can be automatically datumed*

*Fix to a position on the verification rig frame*

*Assign a CAD ID commensurate with that position*

*Make an association between the position and serial number in the verification software interface*

*Manually reprogram each FPU with the latest firmware*

For each of 6 FPUs

Flash the FPU PCB with the correct serial number

Initialise the FPU position database with (-180,0)

Find beta datum

Find alpha datum

Find datum for all FPUs simultaneously

If all FPUs successful

Flag FPU as **initialiseOK**

Start the automated verification tests (as below) 🡪

Else

For each failed FPU

Flag FPU as **initialiseFail**

Stop with an error – FPU or electronics failure ●.

*If any FPUs have failed*

*If datum is confirmed non-functional*

*Remove FPU from verification rig*

*Choose a new FPU to fill this slot*

*Update the serial number to turntable slot to CAN bus mapping*

*Else*

*Manually correct the non-datum issue*

*Flag FPU for retest and leave it on the rig*

*Restart the initialisation process until all FPUs can be datumed and the automated tests can begin.*

## Verification rig setup

Purpose – this is a self-check of the verification rig critical functions, whereby sample images are taken from all test stations and the image analysis functions are run to verify that the images resulting are as expected.

Configurable parameters

* **pupalnPosN** – the rotary stage angle required to place each FPU under the first pupil alignment fold mirror, degrees
* **pupalnLinposN** – the linear stage position required to illuminate each FPU fibre, mm
* **pupalnExposure** – the exposure time for a correctly exposed image, microseconds
* **metcalPosN** - the rotary stage angle required to place each FPU under the metcal camera, degrees
* **metcalExposureTarget** – the exposure time for a correctly exposed image of the targets, microseconds
* **methtPosN** - the rotary stage angle required to place each FPU in front of the metht camera, degrees
* **methtExposure** – the exposure time for a correctly exposed image, microseconds
* **posrepPosN** – the rotary stage angle required to place each FPU under the positional repeatability camera, degrees
* **posrepExposure** – the exposure time for a correctly exposed image, microseconds

Home the turntable and the linear stage

Turn off all lights

Move rotary stage to **pupalnPosN** *(4.2)*

Move linear stage to **pupalnLinposN** *(4.3)*

Configure pupaln camera with **pupalnExposure** *(4.5)*

Configure lighting for pupil alignment test (ambient/silhouette off, backlight 5V) *(4.4)*

Take image with pupaln camera *(4.5)*

Pass image to pupil alignment image analysis script *(5.5)*

If exception returned then

Stop with an error – pupil alignment equipment failure ●

Move rotary stage to **metcalPosN** *(4.2)*

Configure metcal camera with **metcalExposure** *(4.5)*

Configure lighting for metrology calibration target image (backlight/silhouette off, ambient on) *(4.4)*

Take image with metcal camera *(4.5)*

Pass image to metrology calibration target image analysis script *(5.2)*

If exception returned then

Stop with an error – metrology calibration equipment failure ●

Move rotary stage to **methtPosN** *(4.2)*

Configure metht camera with **methtExposure** *(4.5)*

Configure lighting for metrology height image (backlight/ambient off, silhouette on) *(4.4)*

Take image with metht camera *(4.5)*

Pass image to metrology height image analysis script *(5.4)*

If error returned then

Stop with an error – metrology height equipment failure ●

Move rotary stage to **posrepPosN** *(4.2)*

Configure posrep camera for **posrepExposure** *(4.5)*

Configure lighting for positional repeatability test (silhouette/backlight off, ambient on) *(4.4)*

Take image with posrep camera *(4.5)*

Pass image to positional repeatability image analysis script *(5.1)*

If error returned then

Stop with an error – positional repeatability equipment failure ●

Home the turntable and the linear stage

Turn off all lights

## Collision detection

Purpose – to functionally test the collision detection circuit using a soft contact with the beta arm. This will give confidence that the system is functioning without putting the FPU at risk of unnecessary damage. The test is carried out by moving the rotary stage instead of the FPU. This means that recovery only required the *enableBetaCollisionProtection* command and not a special recovery movement.

Configurable parameters

* **coldetPosN –** the rotary stage angle at which each FPU can touch the soft contact
* **coldetAlpha –** the alpha angle at which the FPU can touch the soft contact when running through **coldetPosN**
* **coldetBeta –** the beta angle at which the FPU can touch the soft contact when running through **coldetPosN**

For each of 6 FPUs

Move rotary stage to (**coldetPosN** – 10°) *(4.2)*

Configure and execute waveform to bring FPU to (**coldetAlpha, coldetBeta**)

Move rotary stage to (**coldetPosN** + 10°) *(4.2)*

If FPU registers collision then

Flag FPU as **coldetOK**

Enable beta collision protection

Run reverseMotion

Run findDatum

Else

Flag FPU as **coldetFail**

Stop with an error – FPU collision detection circuit non-functional ●

## Limit characterisation

Purpose – to determine where the end stop limits of the FPU are with respect to the datum position. This will enable the software safety limits to be set for each FPU. This will be achieved by safely activating the limit switch of each stage in both directions.

For the beta arm, this means driving to a position slightly beyond the expected position of the end stop switch in both positive and negative directions. Both tests will require the software to successfully handle exceptions from the FPU driver and then for the beta stop recovery process to be followed.

For the alpha arm, the datum position represents the limit in the negative direction. The positive limit will be found by driving to a positive slightly beyond the expected position of the datum switch. This will require the software to successful handle an exception from the FPU driver and then for the alpha limit breach recovery process to be followed.

Configurable constants



## Pupil alignment

Purpose - to determine precession of the FPU optical axis as it rotates through 16 positions representing all combinations of the major compass directions on both alpha and beta arms. A high-power LED backlights the fibre, which is projected onto a screen via fold mirrors and imaged by a camera. Software finds the centre of the projected spot in the image and, from this, the errors in the chassis, alpha and beta axes can be derived.

Configurable parameters

* **pupalnPosN** – the rotary stage angle required to place each FPU under the first pupil alignment fold mirror, degrees
* **pupalnLinposN** – the linear stage position required to illuminate each FPU fibre, mm
* **pupalnExposure** – the exposure time for a correctly exposed image, microseconds
* **pupalnPass** – the maximum total deviation from the calibrated centre point which represents an acceptable FPU, degrees

Read configurable parameters

Configure pupaln camera with **pupalnExposure** *(4.5)*

Configure lighting for pupil alignment test (ambient/silhouette off, backlight 5V) *(4.4)*

For each of 6 FPUs

Move rotary stage to **pupalnPosN** *(4.2)*

Move linear stage to **pupalnLinposN** *(4.3)*

Configure and execute waveform (+10, -170)

For 4 iterations

Take image with pupaln camera *(4.5)*

For 3 iterations

Configure and execute waveform (0,90)

Take image with pupaln camera *(4.5)*

If Not the final iteration then configure and execute waveform (90,-270)

Configure and execute waveform (-279,-279)

Run findDatum

For 16 saved images

Pass image to pupil alignment image analysis script *(5.5)*

Pass coordinates to pupil alignment parameter script *(6.2)*

Write **pupalnChassisErr, pupalnAlphaErr**, **pupalnBetaErr** and **pupalnTotalErr** to database

If **pupalnTotalErr** is less than **pupalnPass** then

Flag FPU as **pupalnOK**

Else

Flag FPU as **pupalnFail**

## Metrology calibration

Purpose – to determine the static relationship between the metrology targets and the fibre aperture for each FPU, allowing the MOONS metrology position to use the targets to infer the position of the fibre. The required parameters are the distance between the fibre aperture and targets in microns, to a precision of +/- 1 microns. An image will be taken of the FPU with an ambient LED illuminating the targets. A second image will be taken with the fibre backlit. Software will find the positions of the targets using a centre of mass technique, then the position of the fibre using a centroiding algorithm.

Configurable parameters

* **metcalPosN** - the rotary stage angle required to place each FPU under the metcal camera, degrees
* **metcalExposureTarget** – the exposure time for a correctly exposed image of the targets, microseconds
* **metcalExposureFibre** – the exposure time for a correctly exposed image of the backlit fibre, microseconds

Read configurable parameters

For each of 6 FPUs

Move rotary stage to **metcalPosN** *(4.2)*

Configure metcal camera with **metcalExposureTarget** *(4.5)*

Configure lighting for metrology calibration target image (backlight/silhouette off, ambient on) *(4.4)*

Take image with metcal camera *(4.5)*

Pass image to metrology calibration target image analysis script *(5.2)*

Configure metcal camera with **metcalExposureFibre** *(4.5)*

Configure lighting for metrology calibration fibre image (ambient/silhouette off, backlight 0.1V) *(4.4)*

Take image with metcal camera *(4.5)*

Pass image to metrology calibration fibre image analysis script *()*

Pass coordinates to metrology target calibration parameter script *(6.1)*

Write **metcalFibre2LargeTargetDistance**, **metcalFibre2LargeTargetDistance** and **metcalTargetVectorAngle** to database

## Metrology height

Purpose – to determine the height of the metrology targets above the beta arm surface, allowing a correction to be made within the MOONS metrology software pipeline. An LED lights a white surface behind the beta arm such that the arm and targets are silhouetted and the edges can be clearly defined. The silhouette is imaged by a camera and software finds the surface of the beta arm, then the surfaces of the targets, and calculates the height.

Configurable parameters

* **methtPosN** - the rotary stage angle required to place each FPU in front of the metht camera, degrees
* **methtExposure** – the exposure time for a correctly exposed image, microseconds
* **methtHeightTolerance** – maximum allowable height of both targets, mm

Read configurable parameters

Configure metht camera with **methtExposure** *(4.5)*

Configure lighting for metrology height image (backlight/ambient off, silhouette on) *(4.4)*

For each of 6 FPUs

Move rotary stage to **methtPosN** *(4.1)*

Take image with metht camera *(4.5)*

Pass image to metrology height image analysis script *(5.4)*

Write **methtSmallTargetHeight** and **methtLargeTargetHeight** to database

If **methtSmallTargetHeight** or **methtLargeTargetHeight** greater than zero and less than **methtHeightTolerance** then

Flag FPU as **methtOK**

Else

Flag FPU as **methtFail**

## Datum repeatability

Purpose – to determine the repeatability of the FPU datum position under repeated datum operations, both with and without a prior FPU motion. An ambient LED illuminates the metrology targets and a camera images the targets after each FPU datum. Software finds the target coordinates using a centre of mass technique, then calculates the deviation from the average position. The maximum and standard deviation of the errors found represent the datum repeatability acceptance criteria.

Configurable constants

* **posrepPosN** – the rotary stage angle required to place each FPU under the posrep camera, degrees
* **posrepExposure** – the exposure time for a correctly exposed image, microseconds
* **datrepIterations** – the number of datum operations made for each test
* **datrepMaxPass** – the acceptable maximum single error from the average position, mm
* **datrepStdPass** – the acceptable standard deviation of errors from the average position, mm

Read configurable parameters

Configure posrep camera for **posrepExposure** *(4.5)*

Configure lighting for positional repeatability test (silhouette/backlight off, ambient on) *(4.4)*

For each of 6 FPUs

Move rotary stage to **posrepPosN** *(4.2)*

For **datrepIterations** iterations (datum)

Run findDatum on FPU

Take image with posrep camera *(4.5)*

Pass image to positional repeatability image analysis script *(5.1)*

For **datrepIterations** iterations (move-then-datum)

Configure and execute waveform (+30,+30)

Run reverseMotion

Run findDatum on FPU

Take image with posrep camera *(4.5)*

Pass image to positional repeatability image analysis script *(5.1)*

Pass coordinates to datum repeatability parameter script *(6.2)*

Write **datrepDatOnlyMax, datrepDatOnlyStd, datrepMoveDatMax** and **datrepMoveDatStd** to database

If **datrepDatOnlyMax** is below **datrepMaxPass**

and **datrepDatOnlyStd** is below **datrepStdPass**

and **datrepMoveDatMax** is below **datrepMaxPass**

and **datrepMoveDatStd** is below **datrepStdPass** then

Flag FPU as **datrepOK**

Else

Flag FPU as **datrepFail**

## Positional repeatability and gearbox calibration

Purpose - to run each FPU stage through a sequence of equally spaced movements, then back, and repeats several times. An ambient LED illuminates the metrology targets and a camera images the targets after each movement. Software finds the target coordinates using a centre of mass technique, then calculates the deviation from the nominal position, as well as the spread of points at that nominal position. From the spread of points at each nominal position, the positional repeatability can be derived, which represents a pass/fail parameter. Also, the error from the nominal position can be converted into a circumferential and radial error, the former of which can be converted into a function which can correct the FPUs movement to improve absolute accuracy, given an acceptable repeatability.

Configurable parameters

* **posrepPosN** – the rotary stage angle required to place each FPU under the positional repeatability camera, degrees
* **posrepExposure** – the exposure time for a correctly exposed image, microseconds
* **posrepPass** – the maximum acceptable deviation from an average position of a grouping of measured points at a given nominal position, mm
* **posrepIncrements** – the number of movements made within each positive sweep from the starting position
* **posrepIterations** – the number of times each FPU sweeps back and forth

Read configurable parameters

Configure posrep camera for **posrepExposure** *(4.5)*

Configure lighting for positional repeatability test (silhouette/backlight off, ambient on) *(4.4)*

For each of 6 FPUs

If **datrepFail** or **pupalnFail** then

Skip FPU

Move rotary stage to **posrepPosN** *(4.2)*

Configure and execute waveform (+10, -170)

Take image with posrep camera *(4.5)*

For **posrepIterations** iterations

For **posrepIncrements** iterations

Configure and execute waveform (+ [320 / **posrepIncrements**], 0)

Take image with posrep camera *(4.5)*

For **posrepIncrements** iterations

Configure and execute waveform (- [320 / **posrepIncrements**], 0)

Take image with posrep camera *(4.5)*

For **posrepIncrements** iterations

Configure and execute waveform (0, + [320 / **posrepIncrements**])

Take image with posrep camera *(4.5)*

For **posrepIncrements** iterations

Configure and execute waveform (0, - [320 / **posrepIncrements**])

Take image with posrep camera *(4.5)*

For all images

Pass image to positional repeatability image analysis script *(5.1)*

Pass coordinates to positional repeatability parameter script *(6.4)*

Write **posrepRSS, posrepAlphaMax, posrepBetaMax** and all **posrepAlphaMaxAtAngleΦ** / **posrepBetaMaxAtAngleΦ** values to database

If **posrepRSS** is less than **posrepPass** then

Flag FPU as **posrepOK**

Else

Flag FPU as **posrepFail**

Pass coordinates to gearbox calibration parameter script *(6.5)*

Write **gearcorAlpha** and **gearcorBeta** to database

## Positional verification

Purpose – to verify that the gearbox calibrations derived in the previous step result in absolute accuracy within specification.

Configurable parameters

* **posrepPosN** – the rotary stage angle required to place each FPU under the positional repeatability camera
* **posrepExposure** – the exposure time in microseconds for a correctly exposed image
* **posverPass** – the maximum deviation from an average position of a grouping of measured points at a given nominal position which represents an acceptable FPU
* **posverIterations** – the number of random positions where real position is tested against nominal position

Read configurable parameters

Configure posrep camera for **posrepExposure** *(4.5)*

Configure lighting for positional repeatability test (silhouette/backlight off, ambient on) *(4.4)*

For each of 6 FPUs

If **datrepFail** or **pupalnFail** or **posrepFail** then

Skip FPU

Move rotary stage to **posrepPosN** *(4.2)*

Apply **gearcorAlpha** and **gearcorBeta** corrections

Configure and execute waveform (10, -170)

Take image with posrep camera *(4.5)*

For 7 iterations

Configure and execute waveform (45,0)

Take image with posrep camera *(4.5)*

For **posverIterations** iterations

Generate new **posrepAlphaRandN** and **posrepBetaRandN**

Write **posrepAlphaRandN** and **posrepBetaRandN** to database

Configure and execute waveform (**posrepAlphaRandN**, **posrepBetaRandN**)

Take image with posrep camera *(4.5)*

For all images

Pass image to positional repeatability image analysis script *(5.1)*

Pass coordinates to positional accuracy parameter script *(6.6)*

Write**posverErrorMax** and all **posverErrorN** valuesto database

If **posverErrorMax** is below **posverPass** then

Flag FPU as **posverOK**

Else

Flag FPU as **posverFail**

# Hardware control specification

## FPU control

* See FPU protocol 2 documentation

## Rotary stage

* Purpose – the rotary stage rotates the verification frame such that FPUs can be positioned under various test stations.
* Rotary stage - Thorlabs NR360S/M - <https://www.thorlabs.de/thorproduct.cfm?partnumber=NR360S/M>
* Controller - Thorlabs BSC201 – <https://www.thorlabs.de/thorproduct.cfm?partnumber=BSC201>
* Required operations
  + Home
  + Forward absolute movement
  + Reverse absolute movement
* Adjustable parameters
  + Speed

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Stage angles | Frame position | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 |
| metcal | 251 | 311 | 11 | 71 | 131 | 191 |
| metht | 268 | 328 | 28 | 88 | 148 | 208 |
| posrep | 132 | 192 | 252 | 312 | 12 | 72 |
| pupaln |  |  |  |  |  |  |

## Linear stage

* Purpose – the linear stage moves a backlight LED and lens such that light is shone into one of the 6 fibre patch cables coupled with the FPU fibres. This allows imaging of the fibre centre and back projection of the fibre aperture such that pupil alignment can be measured.
* Linear stage - Thorlabs MTS50/M-Z8 - <https://www.thorlabs.de/newgrouppage9.cfm?objectgroup_id=3002&pn=MTS50/M-Z8#3006>
* Controller - Thorlabs BSC201 – <https://www.thorlabs.de/thorproduct.cfm?partnumber=BSC201>
* Required operations
  + Home
  + Forward absolute movement
  + Reverse absolute movement
* Adjustable parameters
  + Speed

## Lamps and control DAQ

* Purpose – three LED lamps allow different setups of ambient and directional illumination within the verification enclosure, as well as backlighting of the fibres.
* Ambient illumination - Thorlabs LED cluster LIU365A - <https://www.thorlabs.de/newgrouppage9.cfm?objectgroup_id=2853>
* Metrology target silhouetting - Thorlabs LED cluster LIU365A - <https://www.thorlabs.de/newgrouppage9.cfm?objectgroup_id=2853>
* Fibre backlight - Thorlabs mounted LED MCWHLP1 - <https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=2692>
* LED driver – Thorlabs LEDD1B - <https://www.thorlabs.de/newgrouppage9.cfm?objectgroup_id=2616>
* DAQ - RedLab1208LS - <https://www.meilhaus.org/downloadserver/redlab/manual/RedLab%201208LS_en.pdf>
* Required operations
  + On-off for ambient illumination and target silhouetting lamps
  + Voltage control (0-5V) for fibre backlight
  + Delay to allow for warmup

## Camera control

* Purpose – four cameras take images of the FPU and fibre projection during the various tests
* Positional repeatability – Basler ace acA3800-10gm GigE - <https://www.edmundoptics.com/p/Basler-ace-acA3800-10gm-Monochrome-GigE-Camera/32412/>
* Metrology calibration – Basler ace acA3800-10gm GigE - <https://www.edmundoptics.com/p/Basler-ace-acA3800-10gm-Monochrome-GigE-Camera/32412/>
* Metrology height – Basler ace acA3800-10gm GigE - <https://www.edmundoptics.com/p/Basler-ace-acA3800-10gm-Monochrome-GigE-Camera/32412/>
* Pupil alignment – Basler ace acA1920-40gm GigE - <https://www.edmundoptics.com/p/basler-ace-aca1920-40gm-monochrome-gige-camera/3429/>
* Required operations
  + Connect to camera and load configuration using given parameters
  + Save image from a single camera given its IP address
* Adjustable parameters
  + Exposure time

|  |  |  |
| --- | --- | --- |
| Camera | ID number | IP address |
| metcal | 22390460 | 169.254.189.121 |
| metht | 22390461 | 169.254.190.121 |
| posrep | 22390458 | 169.254.187.121 |
| pupaln | 22584939 | 169.254.108.113 |

# Image analysis specification

## Find metrology targets from posrep camera

* Purpose – to find coordinates for the metrology targets in real space using the posrep camera. Required accuracy is < 0.01 mm.
* Used in the following tests:
  + Datum repeatability *(3.9)*
  + Positional repeatability and gearbox correction *(3.10)*
  + Positional verification *(3.11)*
* Inputs
  + Path - bitmap image on file from posrep camera
* Configurable parameters
  + **posrepSmallDiameter** – expected diameter of small target, mm
  + **posrepLargeDiameter** – expected diameter of large target, mm
  + **posrepDiametertolerance** – allowable tolerance of both targets, mm
  + **posrepQualityMetric** – minimum circularity for target identification, dimensionless
  + **posrepThreshold** – pixel intensity for image thresholding, 0-255
  + **posrepPlatescale** – image platescale, mm/pixel
  + **posrepDistortionMatrix** – camera calibration matrix
* Return
  + **posrepSmallTargetX** – X coordinate of small target, mm
  + **posrepSmallTargetY** – Y coordinate of small target, mm
  + **posrepSmallTargetQuality** – circularity of small target contour, dimensionless
  + **posrepLargeTargetX** – X coordinate of large target, mm
  + **posrepLargeTargetY** – Y coordinate of large target, mm
  + **posrepLargeTargetQuality** – circularity of large target contour, dimensionless
* Exceptions
  + Multiple small targets found
  + Multiple large targets found
  + No small targets found
  + No large targets found
* Algorithm
  + Process image (distortion correction, grayscale, Gaussian blur, threshold)
  + Find contours
  + Filter contours on size and circularity to detect targets and reject contamination
  + Find centre of targeted contours using Centre of Mass

## Find metrology targets from metcal camera

* Purpose – to find coordinates for the metrology targets in real space using the metcal camera. Required accuracy is < 0.002 mm.
* Used in the following tests:
  + Metrology calibration *(3.7)*
* Inputs
  + Path - bitmap image on file from metcal camera
* Configurable parameters
  + **metcalSmallDiameter** – expected diameter of small target, mm
  + **metcalLargeDiameter** – expected diameter of large target, mm
  + **metcalDiametertolerance** – allowable tolerance of both targets, mm
  + **metcalQualityMetric** – minimum circularity for target identification, dimensionless
  + **metcalThreshold** – pixel intensity for image thresholding, 0-255
  + **metcalPlatescale** – image platescale, mm/pixel
* Return
  + **metcalSmallTargetX** – X coordinate of small target, mm
  + **metcalSmallTargetY** – Y coordinate of small target, mm
  + **metcalSmallTargetQuality** – circularity of small target contour, dimensionless
  + **metcalLargeTargetX** – X coordinate of large target, mm
  + **metcalLargeTargetY** – Y coordinate of large target, mm
  + **metcalLargeTargetQuality** – circularity of large target contour, dimensionless
* Exceptions
  + Multiple small targets found
  + Multiple large targets found
  + No small targets found
  + No large targets found
* Algorithm
  + Process image (grayscale, Gaussian blur, threshold)
  + Find contours
  + Filter contours on size and circularity to detect targets and reject contamination
  + Find centre of targeted contours using Centre of Mass

## Find backlit fibre from metcal camera

* Purpose – to find coordinates for the metrology targets in real space using the metcal camera. Required accuracy is < 0.002 mm.
* Used in the following tests:
  + Metrology calibration *(3.7)*
* Inputs
  + Path - bitmap image on file from metcal camera
* Configurable parameters
  + **metcalPlatescale** – image platescale, mm/pixel
  + **metcalTBD** – unknown parameters relating to Gaussian fit
* Return
  + **metcalFibreX** – X coordinate of fibre, mm
  + **metcalFibreY** – Y coordinate of fibre, mm
  + **metcalTBD** – unknown parameter relating to quality of Gaussian fit
* Exceptions
  + Unable to find maxima
  + Unable to fit Gaussian distribution
  + Unacceptable quality of fit
* Algorithm
  + Find fibre location (find maxima)
  + Find Gaussian fit around point

## Find metrology target heights from metht camera

* Purpose – to measure the height of the metrology targets above the beta arm surface to allow correction for elongation in the metrology system. Required accuracy is < 0.01 mm.
* Used in the following tests:
  + Metrology height *(3.8)*
* Inputs
  + Path - bitmap image on file from metht camera
* Configurable parameters
  + **methtPlatescale** – image platescale, mm/pixel
  + **methtThreshold** – pixel intensity for image thresholding, 0-255
  + **methtScanHeight** – height at which image is scanned to find side of beta arm, pixels
  + **methtGaussBlur** – Gaussian blur parameter, pixels (must be odd number)
  + **methtStandardDev** – acceptable standard deviation of target points, mm
  + **methtNoiseMetric** – acceptable image noise metric, dimensionless
* Return
  + **methtSmallTargetHeight** – height of small target above beta arm, mm
  + **methtLargeTargetHeight** – height of large target above beta arm, mm
  + **methtSmallTargetQuality** – standard deviation of small target points, mm
  + **methtLargeTargetQuality** – standard deviation of large target points, mm
* Exceptions
  + Standard deviation of small target points exceeds limit
  + Standard deviation of large target points exceeds limit
  + Excessive image noise
* Algorithm
  + Process image (Gaussian blur, grayscale and threshold)
  + Check image noise
  + Find side of beta arm
  + At fixed offsets from arm side, find transition pixel representing top edge of arm and targets
  + Fit line through arm points
  + Find normal from line to target points
  + Average normals to give target height

## Find projected spot from pupaln camera

* Purpose – to measure misalignment of the projected axis from a backlit fibre as the FPU rotates and thus determine the angular errors in each of the three mechanism axes. Required accuracy is < 1 mm.
* Used in the following tests:
  + Pupil alignment *(3.6)*
* Inputs
  + Path - bitmap image on file from pupaln camera
* Configurable parameters
  + **pupalnCircularityThresh** – minimum circularity for target identification, dimensionless
  + **pupalnThreshold** – pixel intensity for image thresholding, 0-255
  + **pupalnNoiseMetric** – acceptable image noise metric, dimensionless
  + **pupalnDistortionMatrix** – camera calibration matrix
  + **pupalnPlatescale** – image platescale, mm/pixel
* Return
  + **pupalnSpotX** – X coordinate of projected spot centre of mass, mm
  + **pupalnSpotY** – Y coordinate of projected spot centre of mass, mm
  + **pupalnCircularity** – circularity of projected spot, dimensionless
* Exceptions
  + Unable to find spot
  + Unacceptable spot quality
  + Unacceptable image noise
* Algorithm
  + Process image (distortion correct, convert, de-noise and threshold)
  + Find contours
  + Filter contours on size and circularity to detect targets and reject contamination
  + Find centre of targeted contour using Centre of Mass

# Performance parameter calculations

## Metrology target calibration

* Input
  + **metcalSmallTargetX** – X coordinate of small target, mm
  + **metcalSmallTargetY** – Y coordinate of small target, mm
  + **metcalLargeTargetX** – X coordinate of large target, mm
  + **metcalLargeTargetY** – Y coordinate of large target, mm
  + **metcalFibreX** – X coordinate of fibre, mm
  + **metcalFibreY** – Y coordinate of fibre, mm
* Return
  + **metcalFibre2LargeTargetDistance** – distance from fibre to large target, mm
  + **metcalFibre2SmallTargetDistance** – distance from fibre to small target, mm
  + **metcalTargetVectorAngle** – angle subtended between vectors from fibre to large & small targets
* Algorithm
  + TBD

## Pupil alignment

* Input
  + **pupalnSpotX** – X coordinate of projected spot centre of mass, mm
  + **pupalnSpotY** – Y coordinate of projected spot centre of mass, mm
* Return
  + **pupalnChassisErr** – chassis angular error, degrees
  + **pupalnAlphaErr** – alpha angular error, degrees
  + **pupalnBetaErr** – beta angular error, degrees
  + **pupalnTotalErr** – total angular error, degrees
* Algorithm
  + TBD

## Datum repeatability

* Input
  + **posrepSmallTargetX** – X coordinate of small target, mm
  + **posrepSmallTargetY** – Y coordinate of small target, mm
  + **posrepLargeTargetX** – X coordinate of large target, mm
  + **posrepLargeTargetY** – Y coordinate of large target, mm
* Return
  + **datrepDatOnlyMax** – maximum error of ‘datum-only’ error, mm
  + **datrepDatOnlyStd** – standard deviation of ‘datum-only’ errors, mm
  + **datrepMoveDatMax** – maximum error of ‘move-then-datum’ error, mm
  + **datrepMoveDaStd** – standard deviation of ‘move-then-datum’ errors, mm
* Algorithm
  + Average all coordinates to create baseline coordinate
  + For each ‘datum-only’ coordinate
    - Subtract baseline coordinate from coordinate
  + For each move-then-datum coordinate
    - Subtract baseline coordinate from move-then-datum coordinate
  + Find standard deviation and maximum ‘error – datum only’ values
  + Find standard deviation and maximum ‘error – move-then-datum’ values

## Positional repeatability

* Input
  + **posrepIncrements** – the number of movements made within each positive sweep from the starting position
  + **posrepSmallTargetX** – X coordinate of small target, mm
  + **posrepSmallTargetY** – Y coordinate of small target, mm
  + **posrepLargeTargetX** – X coordinate of large target, mm
  + **posrepLargeTargetY** – Y coordinate of large target, mm
* Return
  + **posrepAlphaMaxAtAngleΦ** – maximum positional error at average of all points at alpha angle Φ, where Φ depends on the configurable parameters in 3.10
  + **posrepBetaMaxAtAngleΦ** – maximum positional error at average of all points at beta angle Φ, where Φ depends on the configurable parameters in 3.10
  + **posrepAlphaMax** – maximum alpha positional error at any angle
  + **posrepBetaMax** – maximum beta positional error at any angle
  + **posrepRSS** – RSS of **posrepAlphaMax** and **posrepBetaMax**
* Algorithm
  + TBD

## Gearbox correction calibration

* Input
  + **posrepIncrements** – the number of movements made within each positive sweep from the starting position
  + **posrepSmallTargetX** – X coordinate of small target, mm
  + **posrepSmallTargetY** – Y coordinate of small target, mm
  + **posrepLargeTargetX** – X coordinate of large target, mm
  + **posrepLargeTargetY** – Y coordinate of large target, mm
* Return
  + **gearcorAlpha** – alpha gearbox correction lookup table / function
  + **gearcorBeta** – beta gearbox correction lookup table / function
* Algorithm
  + TBD

## Positional accuracy

* Input
  + **posverAlphaRandN** – Randomized alpha positions
  + **posverBetaRandN** – Randomized beta positions
  + **posrepSmallTargetX** – X coordinate of small target, mm
  + **posrepSmallTargetY** – Y coordinate of small target, mm
  + **posrepLargeTargetX** – X coordinate of large target, mm
  + **posrepLargeTargetY** – Y coordinate of large target, mm
* Return
  + **posverErrorN** – Positional error against nominal position, mm
  + **posverErrorMax** – Maximum positional error against nominal position, mm
* Algorithm
  + TBD

# Database

DATE/XXXX/TEST/1/

XXXX\_TEST\_N\_001