RCCM

0.98

Generated by Doxygen 1.8.13

Contents

1 Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

2 Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

3 Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

4 Namespace Documentation

4.1 CameraCalibration Namespace Reference

Classes

- class CameraCalibration

  Plugin for calibrating camera by image processing technique
- class CameraCalibrationPlugin

  Defines parameters for automatic calibration plugin

4.2 PressureCameraTrigger Namespace Reference

Classes

- class PressureCameraTrigger

  Captures images when triggger pressure is reached
- class PressureCameraTriggerPlugin

4.3 RCCM Namespace Reference

Namespaces

Classes

- class CycleCounter

  Tracks the cycle number of the test. Can be paused and restarted. This is a virtual implementation of ICycleCounter
- class DataqCycleCounter

  Cycle counter using analog pressure input to get pressure and cycle
- interface ICamera

  Interface defining the basic functionalities of the NFOV and WFOV cameras to allow both cameras to be referenced by the exact same code
- interface ICycleCounter
- interface IRCCMPlugin

  Interface that plugins must implement to provide identifying information about plugin
- interface IRCCMPluginActor

  Interface that allows plugins to define a function that is called to run them
- class Logger

  Helper object for logging debug information with relevant metadata
- class Measurement

  Class representing a measurement of a crack vertex and RCCM data on when it was taken
- class MeasurementSequence
- class Motor

  Abstract representation of stepper motor. Defines the minimum functions and variables needed to define the motor
- class NFOV

  Class that handles connecting to and operating NFOV camera (BlackFly
- class NFOVLensController

  Class for operating Gardasoft controller and focusing both optotune liquid lenses
- class PanelView

  Object used to draw panel graphic to show location of stages
- class **Program**

  Main program
- class RCCMPluginLoader

Plugin loading framework from https://code.msdn.microsoft.com/windowsdesktop/Creating-a-simple-plugi

- class RCCMSystem

  Object representing all the hardware and definitions for the RCCM
- class Settings

  Object representing settings json file
- class TestResults

  Helper object for plotting test status such as crack lengths and pressure
- class TrioController

  Class representing Trio stepper motor controller
- class TrioStepperMotor

  Object representing a physical motor controlled through Trio controller
- class TrioStepperZMotor

  Actuator controlled through trio controller that adjusts its position based on distance sensor input
- class VirtualMotor

  Virtual representation of motor for use when motor is not connected
- class WFOV

  Class representing DMK Z12G445 camera for the RCCM WFOV

Enumerations

- enum RCCMStage { **RCCM1**, **RCCM2**, **Coarse**, **None** }

  Enum representing the two sets of fine axes
- enum MeasurementMode { **Projection**, **Tip**, **Total** }

  Enum representing different methods for calculating crack length
- enum CoordinateSystem { **Global**, **Local** }

  Enum representing different global (FASTER facility) vs local (pabel) coordinate system

4.3.1 Enumeration Type Documentation

4.3.1.1 CoordinateSystem

```
enum RCCM.CoordinateSystem  [strong]
```

Enum representing different global (FASTER facility) vs local (pabel) coordinate system

4.3.1.2 MeasurementMode

```
enum RCCM.MeasurementMode  [strong]
```

Enum representing different methods for calculating crack length

4.3.1.3 RCCMStage

```
enum RCCM.RCCMStage  [strong]
```

Enum representing the two sets of fine axes

4.4 RCCM.Properties Namespace Reference

Classes

- class **Resources**

  A strongly-typed resource class, for looking up localized strings, etc.
- class **Settings**

4.5 RCCM.UI Namespace Reference

Classes

- class AboutRCCMForm

  Form for displaying info about program and RCCM team
- class CameraSettingsForm

  Form for adjusting some camera settings
- class CoordinateSystemSettingsForm

  Form for modifying coordinate system settings such as offsets and angles
- class LensCalibrationForm

  GUI for adjusting NFOV lens focus calibration
- class MotorSettingsForm

  Form for adjusting settings of individual actuators
- class NewMeasurementForm

  Form for defining settings for a new MeasurementSequence
- class NFOVViewForm

  Form for displaying NFOV live image and measurement overlay
- class PluginInitializationForm

  Form for starting plugin and entering inputs
- class RCCMMainForm

  The main window of the program from which all hardware initialization and termination is directed
- class WFOVViewForm

  Form for displaying WFOV live image and measurement overlay

4.6 RepeatabilityTest Namespace Reference

Classes

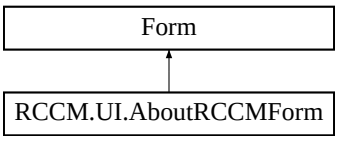
- class RepeatabilityTest
- class RepeatabilityTestPlugin

5 Class Documentation

5.1 RCCM.UI.AboutRCCMForm Class Reference

Form for displaying info about program and RCCM team

Inheritance diagram for RCCM.UI.AboutRCCMForm:



Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

5.1.1 Detailed Description

Form for displaying info about program and RCCM team

5.1.2 Member Function Documentation

5.1.2.1 Dispose()

```
override void RCCM.UI.AboutRCCMForm.Dispose (
            bool disposing )  [protected]
```

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

The documentation for this class was generated from the following files:

- RCCM/UI/AboutRCCMForm.cs
- RCCM/UI/AboutRCCMForm.Designer.cs

5.2 RCCM.UI.LensCalibrationForm.CalibrationPoint Class Reference

Value class for storing the input output pairs of a calibration point

Public Member Functions

- **CalibrationPoint** (double i, double f)

Properties

- double **InputPower** `[get, set]`
- double **FocalPower** `[get, set]`

5.2.1 Detailed Description

Value class for storing the input output pairs of a calibration point

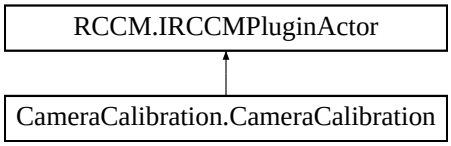
The documentation for this class was generated from the following file:

- RCCM/UI/LensCalibrationForm.cs

5.3 CameraCalibration.CameraCalibration Class Reference

Plugin for calibrating camera by image processing technique

Inheritance diagram for CameraCalibration.CameraCalibration:



Public Member Functions

- CameraCalibration (RCCMSystem rccm, Dictionary< string, string > parameters)
  Create plugin actor from parameter strings
- void Run ()
  Moves actuators and captures images for calibration
- void Stop ()

Static Public Member Functions

- static double **SumSquaresResidual** (double[ ] x, double[,] y, double m, double b)
- static double SumSquaresTotal (double[,] y)
  Calculate total error from mean
- static double FindLinearLeastSquaresFit (double[ ] x, double[,] y, out double m, out double b)
  Find the least squares linear fit.

Protected Attributes

- readonly RCCMSystem **rccm**
- readonly ICamera **camera**
- string **path**
- readonly Motor **xMotor**
- readonly Motor **yMotor**
- Thread **testThread**

Properties

- bool **Running** `[get]`

5.3.1 Detailed Description

Plugin for calibrating camera by image processing technique

5.3.2 Constructor & Destructor Documentation

5.3.2.1 CameraCalibration()

```
CameraCalibration.CameraCalibration.CameraCalibration (
            RCCMSystem rccm,
            Dictionary< string, string > parameters )
```

Create plugin actor from parameter strings

Parameters

rccm	Reference to RCCM object
parameters	Map of test parameters to values

5.3.3 Member Function Documentation

5.3.3.1 FindLinearLeastSquaresFit()

```
static double CameraCalibration.CameraCalibration.FindLinearLeastSquaresFit (
            double [] x,
            double y[,],
            out double m,
            out double b )  [static]
```

Find the least squares linear fit.

Parameters

x	x values of fit data points
y	y values of fit data points
m	slope of best fit line
b	y-intercept

Returns

Total error from best fit line

5.3.3.2 Run()

```
void CameraCalibration.CameraCalibration.Run ( )
```

Moves actuators and captures images for calibration

summary> Stops the test by interrupting the test thread /summary>

Implements RCCM.IRCCMPluginActor.

5.3.3.3 Stop()

```
void CameraCalibration.CameraCalibration.Stop ( )
```

summary> Return the error squared. /summary>

Implements RCCM.IRCCMPluginActor.

5.3.3.4 SumSquaresTotal()

```
static double CameraCalibration.CameraCalibration.SumSquaresTotal (
            double y[,] )  [static]
```

Calculate total error from mean

Parameters

y	y values of linear fit

Returns

Total error from mean

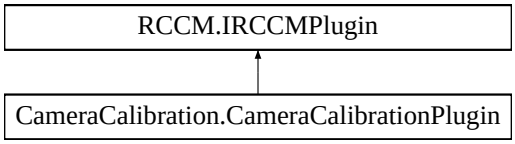
The documentation for this class was generated from the following file:

- RCCM/Plugins/CameraCalibration/CameraCalibration.cs

5.4 CameraCalibration.CameraCalibrationPlugin Class Reference

Defines parameters for automatic calibration plugin

Inheritance diagram for CameraCalibration.CameraCalibrationPlugin:



Public Member Functions

- IRCCMPluginActor Instance (RCCMSystem rccm, Dictionary< string, string > parameters)
  Create plugin with given test parameters

Properties

- string Name `[get]`
  Publicly visible plugin name
- string [ ] Params `[get]`
  Test has one parameter - name of camera being calibrated

5.4.1 Detailed Description

Defines parameters for automatic calibration plugin

5.4.2 Member Function Documentation

5.4.2.1 Instance()

```
IRCCMPluginActor CameraCalibration.CameraCalibrationPlugin.Instance (
            RCCMSystem rccm,
            Dictionary< string, string > parameters )
```

Create plugin with given test parameters

Parameters

rccm	
parameters	User entered test parameters

Returns

Implements RCCM.IRCCMPlugin.

5.4.3 Property Documentation

5.4.3.1 Name

```
string CameraCalibration.CameraCalibrationPlugin.Name  [get]
```

Publicly visible plugin name

5.4.3.2 Params

```
string [] CameraCalibration.CameraCalibrationPlugin.Params  [get]
```

Test has one parameter - name of camera being calibrated

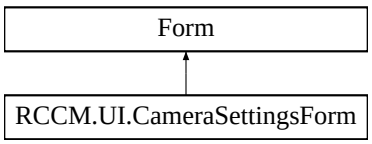
The documentation for this class was generated from the following file:

- RCCM/Plugins/CameraCalibration/CameraCalibrationPlugin.cs

5.5 RCCM.UI.CameraSettingsForm Class Reference

Form for adjusting some camera settings

Inheritance diagram for RCCM.UI.CameraSettingsForm:



Public Member Functions

- CameraSettingsForm (RCCMSystem rccm)

  Create camera settings form

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

Protected Attributes

- readonly RCCMSystem **rccm**

5.5.1 Detailed Description

Form for adjusting some camera settings

5.5.2 Constructor & Destructor Documentation

5.5.2.1 CameraSettingsForm()

```
RCCM.UI.CameraSettingsForm.CameraSettingsForm (
            RCCMSystem rccm )
```

Create camera settings form

Parameters

rccm	Reference to RCCM object

5.5.3 Member Function Documentation

5.5.3.1 Dispose()

```
override void RCCM.UI.CameraSettingsForm.Dispose (
            bool disposing )  [protected]
```

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

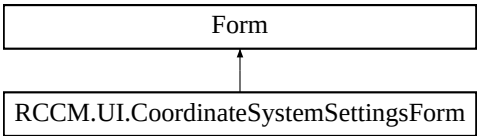
The documentation for this class was generated from the following files:

- RCCM/UI/CameraSettingsForm.cs
- RCCM/UI/CameraSettingsForm.Designer.cs

5.6 RCCM.UI.CoordinateSystemSettingsForm Class Reference

Form for modifying coordinate system settings such as offsets and angles

Inheritance diagram for RCCM.UI.CoordinateSystemSettingsForm:



Public Member Functions

- CoordinateSystemSettingsForm (RCCMSystem rccm)
  Open coordinate system form

Protected Member Functions

- override void Dispose (bool disposing)
  Clean up any resources being used.

Protected Attributes

- readonly RCCMSystem **rccm**

5.6.1 Detailed Description

Form for modifying coordinate system settings such as offsets and angles

5.6.2 Constructor & Destructor Documentation

5.6.2.1 CoordinateSystemSettingsForm()

```
RCCM.UI.CoordinateSystemSettingsForm.CoordinateSystemSettingsForm (
            RCCMSystem rccm )
```

Open coordinate system form

Parameters

rccm	Reference to RCCM object

5.6.3 Member Function Documentation

5.6.3.1 Dispose()

```
override void RCCM.UI.CoordinateSystemSettingsForm.Dispose (
            bool disposing )  [protected]
```

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

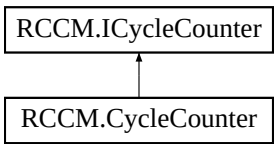
The documentation for this class was generated from the following files:

- RCCM/UI/CoordinateSystemSettingsForm.cs
- RCCM/UI/CoordinateSystemSettingsForm.Designer.cs

5.7 RCCM.CycleCounter Class Reference

Tracks the cycle number of the test. Can be paused and restarted. This is a virtual implementation of ICycleCounter

Inheritance diagram for RCCM.CycleCounter:



Public Member Functions

- CycleCounter (int period)
  Create a cycle counter with a fixed period
- void Start ()
  Activates the counter without resetting the cycle number
- void Start (int cycle)
  Activates the counter without and sets the current cycle to the specified value
- void Stop ()
  Stops the cycle counting. Cycle counting can be resumed
- double GetPressure ()
  Get current pressure reading (simulated with sine curve)
  Returns
  current pressure reading
- int GetElapsed ()
  Elapsed time in test from current cycle and elapsed time since last tick
- async Task Terminate ()
  Implementing a required function that serves no purpose for virtual implementation

Protected Attributes

- Timer countTimer

  Timer for automatically incrementing cycle number
- int period

  Fixed cycle period in milliseconds
- int lastTick

  Environment tick count of last cycle increment

Properties

- bool Active `[get]`

  A boolean indicating whether or not the counter is active
- int Cycle `[get, set]`

  Current cycle number
- int Period `[get, set]`

  Cycle period in milliseconds
- double Amplitude `[get, set]`

  Amplitude of pressure signal

5.7.1 Detailed Description

Tracks the cycle number of the test. Can be paused and restarted. This is a virtual implementation of ICycleCounter

5.7.2 Constructor & Destructor Documentation

5.7.2.1 CycleCounter()

```
RCCM.CycleCounter.CycleCounter (
            int period )
```

Create a cycle counter with a fixed period

Parameters

period	Cycle period in milliseconds

5.7.3 Member Function Documentation

5.7.3.1 GetElapsed()

```
int RCCM.CycleCounter.GetElapsed ( )
```

Elapsed time in test from current cycle and elapsed time since last tick

Implements RCCM.ICycleCounter.

5.7.3.2 GetPressure()

```
double RCCM.CycleCounter.GetPressure ( )
```

Get current pressure reading (simulated with sine curve)

Returns

current pressure reading

Implements RCCM.ICycleCounter.

5.7.3.3 Start() [1/2]

```
void RCCM.CycleCounter.Start ( )
```

Activates the counter without resetting the cycle number

Implements RCCM.ICycleCounter.

5.7.3.4 Start() [2/2]

```
void RCCM.CycleCounter.Start (
            int cycle )
```

Activates the counter without and sets the current cycle to the specified value

Parameters

cycle	Cycle number to start at

Implements RCCM.ICycleCounter.

5.7.3.5 Stop()

```
void RCCM.CycleCounter.Stop ( )
```

Stops the cycle counting. Cycle counting can be resumed

Implements RCCM.ICycleCounter.

5.7.3.6 Terminate()

```
async Task RCCM.CycleCounter.Terminate ( )
```

Implementing a required function that serves no purpose for virtual implementation

Returns

Nothing

Implements RCCM.ICycleCounter.

5.7.4 Member Data Documentation

5.7.4.1 countTimer

```
Timer RCCM.CycleCounter.countTimer  [protected]
```

Timer for automatically incrementing cycle number

5.7.4.2 lastTick

```
int RCCM.CycleCounter.lastTick  [protected]
```

Environment tick count of last cycle increment

5.7.4.3 period

```
int RCCM.CycleCounter.period  [protected]
```

Fixed cycle period in milliseconds

5.7.5 Property Documentation

5.7.5.1 Active

```
bool RCCM.CycleCounter.Active  [get]
```

A boolean indicating whether or not the counter is active

5.7.5.2 Amplitude

```
double RCCM.CycleCounter.Amplitude  [get], [set]
```

Amplitude of pressure signal

5.7.5.3 Cycle

```
int RCCM.CycleCounter.Cycle  [get], [set]
```

Current cycle number

5.7.5.4 Period

```
int RCCM.CycleCounter.Period  [get], [set]
```

Cycle period in milliseconds

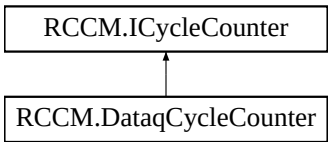
The documentation for this class was generated from the following file:

- RCCM/CycleCounter.cs

5.8 RCCM.DataqCycleCounter Class Reference

Cycle counter using analog pressure input to get pressure and cycle

Inheritance diagram for RCCM.DataqCycleCounter:



Public Member Functions

- DataqCycleCounter ()
  Create dataq device and counter. Note: initialization is asynchronous
- void Start ()
  Activates the counter without resetting the cycle number
- void Start (int cycle)
  Activates the counter without and sets the current cycle to the specified value
- void Stop ()
  Stops the cycle counting. Cycle counting can be resumed
- double GetPressure ()
  Get most recent analog pressure reading
- int GetElapsed ()
  Get time elasped in test in milliseconds
- async Task Terminate ()
  Stop data acquisition and release DATAQ device

Static Public Member Functions

- static double PwlInterp (double[,] data, double val)

  Helper function for piecewise linear interpolation

Static Public Attributes

- static double THRESHOLD = (double)Program.Settings.json["cycle counter"]["threshold"]

  Threshold voltage that must be crossed to interpret as a cycle signal
- static int IN_PRESSURE = (int)Program.Settings.json["cycle counter"]["pressure input"]

  Input channel for pressure reading - a number from 1 to 4
- static int IN_CYCLE = (int)Program.Settings.json["cycle counter"]["cycle input"]

  Input channel for cycle input - a number from 1 to 4

Protected Member Functions

- async void initialize ()

  Asynchronous function that initializes DATAQ device and starts data acquisition
- async void readDataLoop ()

  Worker function for background task that reads data from DATAQ device.

Protected Attributes

- int elapsed

  Accumulator for ticks elapsed in test
- int startTick

  Environment tick count when cycle counting started
- Dataq.Devices.DI1100.Device di_1100

  Handle to DATAQ DI-1100 data acquisition device
- double [,] calibration

  Conversion from volts to pressure. Represented as rows of (voltage, pressure) pairs that are linearly interpolated to get pressure from a given voltage reading
- double pressure

  Stores last pressure reading
- double lastCycleVoltage

  Stores last cycle input reading
- AnalogVoltageIn pressureChannel

  Channel for pressure reading
- AnalogVoltageIn cycleChannel

  Channel for cycle input
- Task readDataTask

  Background task for reading data
- CancellationTokenSource cancelRead

  Token for signalling cancellation to background task

Properties

- bool Active `[get, protected set]`

  A boolean indicating whether or not the counter is active
- int Cycle `[get, set]`

  Current cycle number

5.8.1 Detailed Description

Cycle counter using analog pressure input to get pressure and cycle

5.8.2 Constructor & Destructor Documentation

5.8.2.1 DataqCycleCounter()

```
RCCM.DataqCycleCounter.DataqCycleCounter ( )
```

Create dataq device and counter. Note: initialization is asynchronous

5.8.3 Member Function Documentation

5.8.3.1 GetElapsed()

```
int RCCM.DataqCycleCounter.GetElapsed ( )
```

Get time elasped in test in milliseconds

Returns

Cumulative milliseconds elapsed, accounting for pauses

Implements RCCM.ICycleCounter.

5.8.3.2 GetPressure()

```
double RCCM.DataqCycleCounter.GetPressure ( )
```

Get most recent analog pressure reading

Returns

Last pressure reading recorded

Implements RCCM.ICycleCounter.

5.8.3.3 initialize()

```
async void RCCM.DataqCycleCounter.initialize ( )  [protected]
```

Asynchronous function that initializes DATAQ device and starts data acquisition

5.8.3.4 PwlInterp()

```
static double RCCM.DataqCycleCounter.PwlInterp (
            double data[,],
            double val )  [static]
```

Helper function for piecewise linear interpolation

Parameters

data	2D array of (x, y) value pairs
val	X value to be interpolated

Returns

Interpolated Y value corresponding to input X

5.8.3.5 readDataLoop()

```
async void RCCM.DataqCycleCounter.readDataLoop ( )  [protected]
```

Worker function for background task that reads data from DATAQ device.

5.8.3.6 Start() [1/2]

```
void RCCM.DataqCycleCounter.Start ( )
```

Activates the counter without resetting the cycle number

Implements RCCM.ICycleCounter.

5.8.3.7 Start() [2/2]

```
void RCCM.DataqCycleCounter.Start (
            int cycle )
```

Activates the counter without and sets the current cycle to the specified value

Parameters

cycle	Cycle number to start at

Implements RCCM.ICycleCounter.

5.8.3.8 Stop()

```
void RCCM.DataqCycleCounter.Stop ( )
```

Stops the cycle counting. Cycle counting can be resumed

Implements RCCM.ICycleCounter.

5.8.3.9 Terminate()

```
async Task RCCM.DataqCycleCounter.Terminate ( )
```

Stop data acquisition and release DATAQ device

Implements RCCM.ICycleCounter.

5.8.4 Member Data Documentation

5.8.4.1 calibration

```
double [,] RCCM.DataqCycleCounter.calibration  [protected]
```

Conversion from volts to pressure. Represented as rows of (voltage, pressure) pairs that are linearly interpolated to get pressure from a given voltage reading

5.8.4.2 cancelRead

```
CancellationTokenSource RCCM.DataqCycleCounter.cancelRead  [protected]
```

Token for signalling cancellation to background task

5.8.4.3 cycleChannel

```
AnalogVoltageIn RCCM.DataqCycleCounter.cycleChannel  [protected]
```

Channel for cycle input

5.8.4.4 di_1100

```
Dataq.Devices.DI1100.Device RCCM.DataqCycleCounter.di_1100  [protected]
```

Handle to DATAQ DI-1100 data acquisition device

5.8.4.5 elapsed

```
int RCCM.DataqCycleCounter.elapsed  [protected]
```

Accumulator for ticks elapsed in test

5.8.4.6 IN_CYCLE

```
int RCCM.DataqCycleCounter.IN_CYCLE = (int)Program.Settings.json["cycle counter"]["cycle input"]
[static]
```

Input channel for cycle input - a number from 1 to 4

5.8.4.7 IN_PRESSURE

```
int RCCM.DataqCycleCounter.IN_PRESSURE = (int)Program.Settings.json["cycle counter"]["pressure
input"]  [static]
```

Input channel for pressure reading - a number from 1 to 4

5.8.4.8 lastCycleVoltage

```
double RCCM.DataqCycleCounter.lastCycleVoltage  [protected]
```

Stores last cycle input reading

5.8.4.9 pressure

```
double RCCM.DataqCycleCounter.pressure  [protected]
```

Stores last pressure reading

5.8.4.10 pressureChannel

```
AnalogVoltageIn RCCM.DataqCycleCounter.pressureChannel  [protected]
```

Channel for pressure reading

5.8.4.11 readDataTask

```
Task RCCM.DataqCycleCounter.readDataTask  [protected]
```

Background task for reading data

5.8.4.12 startTick

```
int RCCM.DataqCycleCounter.startTick  [protected]
```

Environment tick count when cycle counting started

5.8.4.13 THRESHOLD

```
double RCCM.DataqCycleCounter.THRESHOLD = (double)Program.Settings.json["cycle counter"]["threshold"]
[static]
```

Threshold voltage that must be crossed to interpret as a cycle signal

5.8.5 Property Documentation

5.8.5.1 Active

```
bool RCCM.DataqCycleCounter.Active  [get], [protected set]
```

A boolean indicating whether or not the counter is active

5.8.5.2 Cycle

```
int RCCM.DataqCycleCounter.Cycle  [get], [set]
```

Current cycle number

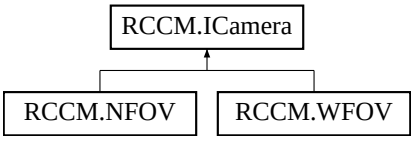
The documentation for this class was generated from the following file:

- RCCM/DataqCycleCounter.cs

5.9 RCCM.ICamera Interface Reference

Interface defining the basic functionalities of the NFOV and WFOV cameras to allow both cameras to be referenced by the exact same code

Inheritance diagram for RCCM.ICamera:



Public Member Functions

- void Start ()

  Begin live image transmission from camera
- void Stop ()

  Stop live image transmission from camera
- void Snap (string filename)

  Save live image from camera to bitmap file
- void Record (string filename)

  Begin recording video from camera
- void StopRecord ()

  Stop recording video from camera
- bool CheckFOV (RCCMSystem rccm)

  Returns true if current state of RCCM matches calibration state for FOV
- void SetScale (RCCMSystem rccm, double scale)

  Set value for pixel to micron conversion and save relevant info for checking that fov is correct

Properties

- double Scale `[get]`

  Conversion from pixels to microns
- double Height `[get]`

  Height in microns of image
- double Width `[get]`

  Width in microns of image
- bool Recording `[get]`

  A flag to indicate if camera is recording video

5.9.1 Detailed Description

Interface defining the basic functionalities of the NFOV and WFOV cameras to allow both cameras to be referenced by the exact same code

5.9.2 Member Function Documentation

5.9.2.1 CheckFOV()

```
bool RCCM.ICamera.CheckFOV (
            RCCMSystem rccm )
```

Returns true if current state of RCCM matches calibration state for FOV

Parameters

rccm	Handle to RCCM object for getting z position

Implemented in RCCM.NFOV, and RCCM.WFOV.

5.9.2.2 Record()

```
void RCCM.ICamera.Record (
            string filename )
```

Begin recording video from camera

Parameters

filename	Path to .avi file where video will save

Implemented in RCCM.NFOV, and RCCM.WFOV.

5.9.2.3 SetScale()

```
void RCCM.ICamera.SetScale (
            RCCMSystem rccm,
            double scale )
```

Set value for pixel to micron conversion and save relevant info for checking that fov is correct

Parameters

rccm	Handle to RCCM object for getting z position
scale	New value of pixels to microns conversion

Implemented in RCCM.NFOV, and RCCM.WFOV.

5.9.2.4 Snap()

```
void RCCM.ICamera.Snap (
            string filename )
```

Save live image from camera to bitmap file

Parameters

filename	Path to .bmp file where image will save

Implemented in RCCM.NFOV, and RCCM.WFOV.

5.9.2.5 Start()

```
void RCCM.ICamera.Start ( )
```

Begin live image transmission from camera

Implemented in RCCM.NFOV, and RCCM.WFOV.

5.9.2.6 Stop()

```
void RCCM.ICamera.Stop ( )
```

Stop live image transmission from camera

Implemented in RCCM.NFOV, and RCCM.WFOV.

5.9.2.7 StopRecord()

```
void RCCM.ICamera.StopRecord ( )
```

Stop recording video from camera

Implemented in RCCM.NFOV, and RCCM.WFOV.

5.9.3 Property Documentation

5.9.3.1 Height

```
double RCCM.ICamera.Height  [get]
```

Height in microns of image

5.9.3.2 Recording

```
bool RCCM.ICamera.Recording  [get]
```

A flag to indicate if camera is recording video

5.9.3.3 Scale

```
double RCCM.ICamera.Scale  [get]
```

Conversion from pixels to microns

5.9.3.4 Width

```
double RCCM.ICamera.Width  [get]
```

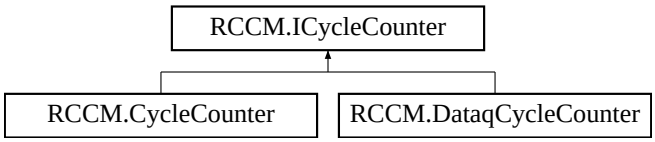
Width in microns of image

The documentation for this interface was generated from the following file:

- RCCM/ICamera.cs

5.10 RCCM.ICycleCounter Interface Reference

Inheritance diagram for RCCM.ICycleCounter:



Public Member Functions

- void Start ()

  Starts cycle counting
- void Start (int cycle)

  Starts cycle counting at the specified cycle number
- void Stop ()

  Stops cycle counting
- double GetPressure ()

  Get most recent pressure reading
- int GetElapsed ()

  Get time elapsed in test
- Task Terminate ()

  Perform any action neccessary for stopping cycle counter

Properties

- bool Active `[get]`

  A boolean indicating whether or not the counter is active
- int Cycle `[get, set]`

  Current cycle number

5.10.1 Member Function Documentation

5.10.1.1 GetElapsed()

```
int RCCM.ICycleCounter.GetElapsed ( )
```

Get time elapsed in test

Returns

milliseconds since test start

Implemented in RCCM.DataqCycleCounter, and RCCM.CycleCounter.

5.10.1.2 GetPressure()

```
double RCCM.ICycleCounter.GetPressure ( )
```

Get most recent pressure reading

Returns

Most recent pressure reading

Implemented in RCCM.DataqCycleCounter, and RCCM.CycleCounter.

5.10.1.3 Start() [1/2]

```
void RCCM.ICycleCounter.Start ( )
```

Starts cycle counting

Implemented in RCCM.DataqCycleCounter, and RCCM.CycleCounter.

5.10.1.4 Start() [2/2]

```
void RCCM.ICycleCounter.Start (
            int cycle )
```

Starts cycle counting at the specified cycle number

Implemented in RCCM.DataqCycleCounter, and RCCM.CycleCounter.

5.10.1.5 Stop()

```
void RCCM.ICycleCounter.Stop ( )
```

Stops cycle counting

Implemented in RCCM.DataqCycleCounter, and RCCM.CycleCounter.

5.10.1.6 Terminate()

```
Task RCCM.ICycleCounter.Terminate ( )
```

Perform any action neccessary for stopping cycle counter

Returns

Task completion

Implemented in RCCM.DataqCycleCounter, and RCCM.CycleCounter.

5.10.2 Property Documentation

5.10.2.1 Active

```
bool RCCM.ICycleCounter.Active  [get]
```

A boolean indicating whether or not the counter is active

5.10.2.2 Cycle

```
int RCCM.ICycleCounter.Cycle  [get], [set]
```

Current cycle number

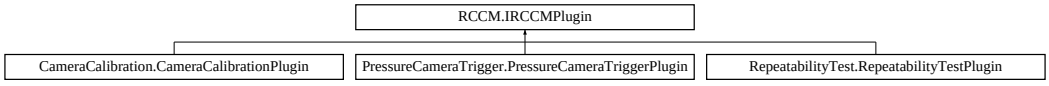
The documentation for this interface was generated from the following file:

- RCCM/ICycleCounter.cs

5.11 RCCM.IRCCMPlugin Interface Reference

Interface that plugins must implement to provide identifying information about plugin

Inheritance diagram for RCCM.IRCCMPlugin:



Public Member Functions

- IRCCMPluginActor Instance (RCCMSystem rccm, Dictionary< string, string > parameters)

  Create an instance of the plugin

Properties

- string Name [get]

  User visible name of plugin
- string [] Params [get]

  Plugin inputs

5.11.1 Detailed Description

Interface that plugins must implement to provide identifying information about plugin

5.11.2 Member Function Documentation

5.11.2.1 Instance()

```
IRCCMPluginActor RCCM.IRCCMPlugin.Instance (
            RCCMSystem rccm,
            Dictionary< string, string > parameters )
```

Create an instance of the plugin

Parameters

rccm	Reference to the RCCM object
parameters	User inputs to the plugin

Returns

An instance of the runnable plugin interface

Implemented in PressureCameraTrigger.PressureCameraTriggerPlugin, CameraCalibration.CameraCalibration↩Plugin, and RepeatabilityTest.RepeatabilityTestPlugin.

5.11.3 Property Documentation

5.11.3.1 Name

```
string RCCM.IRCCMPlugin.Name [get]
```

User visible name of plugin

5.11.3.2 Params

```
string [] RCCM.IRCCMPlugin.Params  [get]
```

Plugin inputs

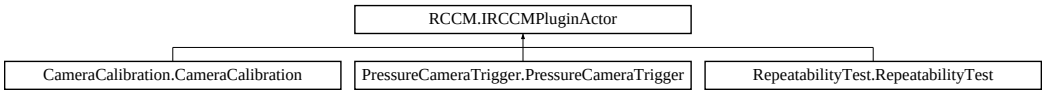
The documentation for this interface was generated from the following file:

- RCCM/IRCCMPlugin.cs

5.12 RCCM.IRCCMPluginActor Interface Reference

Interface that allows plugins to define a function that is called to run them

Inheritance diagram for RCCM.IRCCMPluginActor:



Public Member Functions

- void Run ()

  Run the function with the inputs already passed to the plugin
- void Stop ()

  This function should cause the plugin to stop as soon as possible

Properties

- bool Running `[get]`

  A flag indicating if the plugin is running

5.12.1 Detailed Description

Interface that allows plugins to define a function that is called to run them

5.12.2 Member Function Documentation

5.12.2.1 Run()

```
void RCCM.IRCCMPluginActor.Run ( )
```

Run the function with the inputs already passed to the plugin

Implemented in CameraCalibration.CameraCalibration, RepeatabilityTest.RepeatabilityTest, and PressureCameraTrigger.PressureCameraTrigger.

5.12.2.2 Stop()

```
void RCCM.IRCCMPluginActor.Stop ( )
```

This function should cause the plugin to stop as soon as possible

Implemented in CameraCalibration.CameraCalibration, RepeatabilityTest.RepeatabilityTest, and PressureCameraTrigger.PressureCameraTrigger.

5.12.3 Property Documentation

5.12.3.1 Running

```
bool RCCM.IRCCMPluginActor.Running  [get]
```

A flag indicating if the plugin is running

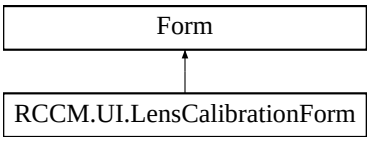
The documentation for this interface was generated from the following file:

- RCCM/IRCCMPluginActor.cs

5.13 RCCM.UI.LensCalibrationForm Class Reference

GUI for adjusting NFOV lens focus calibration

Inheritance diagram for RCCM.UI.LensCalibrationForm:



Classes

- class CalibrationPoint

  Value class for storing the input output pairs of a calibration point

Public Member Functions

- LensCalibrationForm (RCCMSystem rccm, RCCMStage stage)

  Create a calibration UI form for the specified stage. Saves changes to specified settings object

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

Protected Attributes

- RCCMSystem rccm

  The RCCM object
- NFOVLensController controller

  Lens controller
- RCCMStage stage

  Parent stage of NFOV camera to be adjusted
- double [,] oldCalibration

  Calibration data before creation of this form. Reverts calibration if cancel is clicked
- SortedList< double, CalibrationPoint > calibration

  Data of the active calibration that the user is entering

5.13.1 Detailed Description

GUI for adjusting NFOV lens focus calibration

5.13.2 Constructor & Destructor Documentation

5.13.2.1 LensCalibrationForm()

```
RCCM.UI.LensCalibrationForm.LensCalibrationForm (
            RCCMSystem rccm,
            RCCMStage stage )
```

Create a calibration UI form for the specified stage. Saves changes to specified settings object

Parameters

controller	NFOV lens controller
stage	Parent stage of NFOV camera to be adjusted

5.13.3 Member Function Documentation

5.13.3.1 Dispose()

```
override void RCCM.UI.LensCalibrationForm.Dispose (
            bool disposing )  [protected]
```

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

5.13.4 Member Data Documentation

5.13.4.1 calibration

```
SortedList<double, CalibrationPoint> RCCM.UI.LensCalibrationForm.calibration  [protected]
```

Data of the active calibration that the user is entering

5.13.4.2 controller

```
NFOVLensController RCCM.UI.LensCalibrationForm.controller  [protected]
```

Lens controller

5.13.4.3 oldCalibration

```
double [,] RCCM.UI.LensCalibrationForm.oldCalibration  [protected]
```

Calibration data before creation of this form. Reverts calibration if cancel is clicked

5.13.4.4 rccm

```
RCCMSystem RCCM.UI.LensCalibrationForm.rccm  [protected]
```

The RCCM object

5.13.4.5 stage

```
RCCMStage RCCM.UI.LensCalibrationForm.stage  [protected]
```

Parent stage of NFOV camera to be adjusted

The documentation for this class was generated from the following files:

- RCCM/UI/LensCalibrationForm.cs
- RCCM/UI/LensCalibrationForm.Designer.cs

5.14 RCCM.Logger Class Reference

Helper object for logging debug information with relevant metadata

Static Public Member Functions

- static void Out (string str, [CallerFilePath] string path="", [CallerLineNumber] int lineNumber=0, [Caller↩ MemberName] string caller="")

  Write a line to the logfile
- static void Save ()

  Save file to disk

Static Public Attributes

- static StreamWriter Logfile = new StreamWriter("log/output.txt", false)

  Log file where debug info will be saved

5.14.1 Detailed Description

Helper object for logging debug information with relevant metadata

5.14.2 Member Function Documentation

5.14.2.1 Out()

```
static void RCCM.Logger.Out (
            string str,
            [CallerFilePath] string path = "",
            [CallerLineNumber] int lineNumber = 0,
            [CallerMemberName] string caller = "" )  [static]
```

Write a line to the logfile

Parameters

str	Line to write
path	Path of file containing calling function (automatic)
lineNumber	Line number of calling function (automatic)
caller	Calling function (automatic)

5.14.2.2 Save()

```
static void RCCM.Logger.Save ( )  [static]
```

Save file to disk

5.14.3 Member Data Documentation

5.14.3.1 Logfile

```
StreamWriter RCCM.Logger.Logfile = new StreamWriter("log/output.txt", false)  [static]
```

Log file where debug info will be saved

The documentation for this class was generated from the following file:

- RCCM/Logger.cs

5.15 RCCM.Measurement Class Reference

Class representing a measurement of a crack vertex and RCCM data on when it was taken

Public Member Functions

- Measurement (MeasurementSequence crack, RCCMSystem rccm, double pixelX, double pixelY)
  Create a measurement
- string ToCSVString ()
  Create csv line containing all data pertaining to this measurement

Static Public Attributes

- static string [ ] CSV_HEADER = { "Timestamp", "Cycle", "Length", "Pressure", "Panel X", "Panel Y", "Coarse X", "Coarse Y", "Fine X", "Fine Y", "Fine Z", "Height", "Pixel X", "Pixel Y", "Global X", "Global Y", "Filename" }
  Fields for CSV header explaining the ordering of data in measurement file

Properties

- int Cycle `[get]`
  Cycle number when measurement was taken
- double X `[get]`
  Global X coordinate where measurement was taken
- double Y `[get]`
  Global Y coordinate where measurement was taken
- double PanelX `[get]`
  X coordinate in panel coordinate system
- double PanelY `[get]`
  Y coordinate in panel coordinate system
- double CrackLength `[get, set]`
  Length of crack at this measurement
- double **Pressure** `[get, protected set]`
- string **Timestamp** `[get, protected set]`
- double **CoarseX** `[get, protected set]`
- double **CoarseY** `[get, protected set]`
- double **FineX** `[get, protected set]`
- double **FineY** `[get, protected set]`
- double **FineZ** `[get, protected set]`
- double **Height** `[get, protected set]`
- double **PixelX** `[get, protected set]`
- double **PixelY** `[get, protected set]`
- string **Filename** `[get, protected set]`

5.15.1 Detailed Description

Class representing a measurement of a crack vertex and RCCM data on when it was taken

5.15.2 Constructor & Destructor Documentation

5.15.2.1 Measurement()

```
RCCM.Measurement.Measurement (
            MeasurementSequence crack,
            RCCMSystem rccm,
            double pixelX,
            double pixelY )
```

Create a measurement

Parameters

crack	Crack containing this measurement
rccm	Reference to RCCM object
pixelX	Pixel horizontal position within image to measurement
pixelY	Pixel vertical position within image to measurement

5.15.3 Member Function Documentation

5.15.3.1 ToCSVString()

```
string RCCM.Measurement.ToCSVString ( )
```

Create csv line containing all data pertaining to this measurement

Returns

CSV string representing this Measurement

5.15.4 Member Data Documentation

5.15.4.1 CSV_HEADER

```
string [] RCCM.Measurement.CSV_HEADER = { "Timestamp", "Cycle", "Length", "Pressure", "Panel
X", "Panel Y", "Coarse X", "Coarse Y", "Fine X", "Fine Y", "Fine Z", "Height", "Pixel X",
"Pixel Y", "Global X", "Global Y", "Filename" }  [static]
```

Fields for CSV header explaining the ordering of data in measurement file

5.15.5 Property Documentation

5.15.5.1 CrackLength

```
double RCCM.Measurement.CrackLength  [get], [set]
```

Length of crack at this measurement

5.15.5.2 Cycle

```
int RCCM.Measurement.Cycle  [get]
```

Cycle number when measurement was taken

5.15.5.3 PanelX

```
double RCCM.Measurement.PanelX  [get]
```

X coordinate in panel coordinate system

5.15.5.4 PanelY

```
double RCCM.Measurement.PanelY  [get]
```

Y coordinate in panel coordinate system

5.15.5.5 X

```
double RCCM.Measurement.X  [get]
```

Global X coordinate where measurement was taken

5.15.5.6 Y

```
double RCCM.Measurement.Y  [get]
```

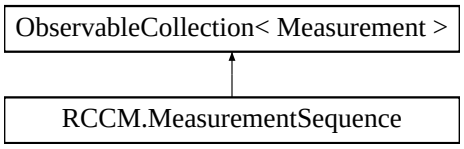
Global Y coordinate where measurement was taken

The documentation for this class was generated from the following file:

- RCCM/Measurement.cs

5.16 RCCM.MeasurementSequence Class Reference

Inheritance diagram for RCCM.MeasurementSequence:



Public Member Functions

- MeasurementSequence (Color color, string name, float size, float orientation, MeasurementMode mode, string camera)

  Create a new measurement sequence with all parameters specified
- MeasurementSequence (NewMeasurementForm parentForm)

  Create the measurement sequence defined by the NewMeasurementSequence form
- void AddPoint (Measurement pt)

  Add a point to the list of vertices in this sequence
- Measurement GetPoint (int ind)

  Get Measurement corresponding to the specified index
- Measurement GetLastPoint ()

  Get last Measurement in sequence
- bool removePoint (int index)

  Deletes a vertex from this sequence
- void Plot (Graphics axes, float scale)

  Plot the line segments of this measurement sequence on a graphics container
- string GetFileName ()

  Create a filename with identifying information about sequence
- bool WriteToFile (string path, bool autoName)

  Write measurement to file. Formatted as one line header followed with a line for each crack vertex
- override string ToString ()

  Return name identifying this sequence
- double CalculateLength (int ind)

  Calculate the length of the crack at the specified measurement
- void RecomputeLength ()

  Recalculate length of crack at time of each measurement

Static Public Attributes

- static int CrackCount = 0

  Counter for number of cracks that have been created so that unique name can be created

Protected Attributes

- List< Measurement > points

  List of points of crack vertices and relevant metadata
- MeasurementMode **mode**

Properties

- string Camera [get, set]

  Indicates which camera captured these measurements
- Color Color [get, set]

  Color of line to display
- string Name [get, set]

  Name of sequence for display
- float LineSize [get, set]

  Width of line
- double Orientation [get, set]

  Angular orientation (in degrees) of initial notch relative to panel coordinate system
- MeasurementMode Mode [get, set]

  Method to use for crack length calculation
- int CountPoints [get]

  Number of points in sequence
- string **CreateTime** [get]
- string **SaveDir** [get]

5.16.1 Constructor & Destructor Documentation

5.16.1.1 MeasurementSequence() [1/2]

```
RCCM.MeasurementSequence.MeasurementSequence (
            Color color,
            string name,
            float size,
            float orientation,
            MeasurementMode mode,
            string camera )
```

Create a new measurement sequence with all parameters specified

Parameters

color	Color of line representation on NFOV image
name	Name identifying crack
size	Thickness of line on NFOV image
orientation	Angular orientation (degrees) of initial crack notch
mode	Crack length calculation method
camera	Camera used to image crack

5.16.1.2 MeasurementSequence() [2/2]

```
RCCM.MeasurementSequence.MeasurementSequence (
            NewMeasurementForm parentForm )
```

Create the measurement sequence defined by the NewMeasurementSequence form

Parameters

parentForm	

5.16.2 Member Function Documentation

5.16.2.1 AddPoint()

```
void RCCM.MeasurementSequence.AddPoint (
            Measurement pt )
```

Add a point to the list of vertices in this sequence

Parameters

pt	Crack vertex

5.16.2.2 CalculateLength()

```
double RCCM.MeasurementSequence.CalculateLength (
            int ind )
```

Calculate the length of the crack at the specified measurement

Parameters

ind	Index in measurement sequence of the desired measurement

Returns

Crack length when specified measurement was made

5.16.2.3 GetFileName()

```
string RCCM.MeasurementSequence.GetFileName ( )
```

Create a filename with identifying information about sequence

Returns

Filename formatted with current timestamp, crack name, and .csv extension

5.16.2.4 GetLastPoint()

```
Measurement RCCM.MeasurementSequence.GetLastPoint ( )
```

Get last Measurement in sequence

Returns

Measurement at last index

5.16.2.5 GetPoint()

```
Measurement RCCM.MeasurementSequence.GetPoint (
            int ind )
```

Get Measurement corresponding to the specified index

Parameters

ind	Index of measurement to return

Returns

Measurement at this index if it is valid

5.16.2.6 Plot()

```
void RCCM.MeasurementSequence.Plot (
            Graphics axes,
            float scale )
```

Plot the line segments of this measurement sequence on a graphics container

Parameters

axes	Graphics object of the container that will display the plot

5.16.2.7 RecomputeLength()

```
void RCCM.MeasurementSequence.RecomputeLength ( )
```

Recalculate length of crack at time of each measurement

5.16.2.8 removePoint()

```
bool RCCM.MeasurementSequence.removePoint (
            int index )
```

Deletes a vertex from this sequence

Parameters

index	Index of the vertex to be deleted

Returns

True or false if deletion was successful

5.16.2.9 ToString()

```
override string RCCM.MeasurementSequence.ToString ( )
```

Return name identifying this sequence

Returns

Sequence name

5.16.2.10 WriteToFile()

```
bool RCCM.MeasurementSequence.WriteToFile (
            string path,
            bool autoName )
```

Write measurement to file. Formatted as one line header followed with a line for each crack vertex

Parameters

path	Filename and directory path where data will be saved
autoName	If true, filename will be automatically defined

Returns

T/F if the write was / wasn't successful

5.16.3 Member Data Documentation

5.16.3.1 CrackCount

```
int RCCM.MeasurementSequence.CrackCount = 0  [static]
```

Counter for number of cracks that have been created so that unique name can be created

5.16.3.2 points

```
List<Measurement> RCCM.MeasurementSequence.points  [protected]
```

List of points of crack vertices and relevant metadata

5.16.4 Property Documentation

5.16.4.1 Camera

```
string RCCM.MeasurementSequence.Camera  [get], [set]
```

Indicates which camera captured these measurements

5.16.4.2 Color

```
Color RCCM.MeasurementSequence.Color  [get], [set]
```

Color of line to display

5.16.4.3 CountPoints

```
int RCCM.MeasurementSequence.CountPoints  [get]
```

Number of points in sequence

5.16.4.4 LineSize

```
float RCCM.MeasurementSequence.LineSize  [get], [set]
```

Width of line

5.16.4.5 Mode

```
MeasurementMode RCCM.MeasurementSequence.Mode  [get], [set]
```

Method to use for crack length calculation

5.16.4.6 Name

```
string RCCM.MeasurementSequence.Name  [get], [set]
```

Name of sequence for display

5.16.4.7 Orientation

```
double RCCM.MeasurementSequence.Orientation  [get], [set]
```

Angular orientation (in degrees) of initial notch relative to panel coordinate system

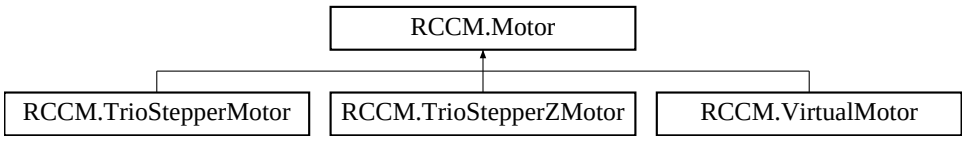
The documentation for this class was generated from the following file:

- RCCM/MeasurementSequence.cs

5.17 RCCM.Motor Class Reference

Abstract representation of stepper motor. Defines the minimum functions and variables needed to define the motor

Inheritance diagram for RCCM.Motor:



Public Member Functions

- Motor ()

  Initialize variables associated with base motor class
- abstract double GetPos ()

  Get current position of actuator
- abstract double SetPos (double cmd)

  Set new command position of actuator
- abstract double MoveRel (double dist)

  Move actuator specified distance from current position
- virtual bool SetProperty (string property, double value)

  Set a motor property
- virtual double GetProperty (string property)

  Gets a motor property value
- void GotoHome ()

  Move to user defined home position
- abstract bool Initialize ()

  Perform any required initialization
- abstract Dictionary< string, double > GetAllProperties ()

  Get all motor settings in a property value pair dictionary
- abstract void WaitForEndOfMove ()

  Blocking function that runs until current actuator move completes
- abstract void Jog (bool fwd)

  Begin continuously moving actuator at jog speed
- abstract void JogStop ()

  Stop jogging
- virtual double GetActuatorPos ()

  Must return actual position of actuator from end of travel. Important for Z actuators
- virtual void Zero ()

  Define current actuator position as zero
- virtual void Terminate ()

  Perform any required action to disconnect from motor

Static Public Attributes

- static string [ ] MOTOR_SETTINGS = { "enabled", "microstep per mm", "velocity", "jog speed", "acceleration", "deceleration", "low position limit", "high position limit", "home", "feedback" }

  List of motor setting names that all motor implementations must include

Protected Attributes

- double commandPos = 0

  Current goal position of actuator
- Dictionary< string, double > settings

  Maintained list of motion settings and limits

Properties

- bool Jogging `[get, protected set]`

  A flag to indicate if the motor is being jogged

5.17.1 Detailed Description

Abstract representation of stepper motor. Defines the minimum functions and variables needed to define the motor

5.17.2 Constructor & Destructor Documentation

5.17.2.1 Motor()

```
RCCM.Motor.Motor ( )
```

Initialize variables associated with base motor class

5.17.3 Member Function Documentation

5.17.3.1 GetActuatorPos()

```
virtual double RCCM.Motor.GetActuatorPos ( )  [virtual]
```

Must return actual position of actuator from end of travel. Important for Z actuators

Returns

Position of actuator from end of travel

Reimplemented in RCCM.TrioStepperZMotor.

5.17.3.2 GetAllProperties()

```
abstract Dictionary<string, double> RCCM.Motor.GetAllProperties ( )  [pure virtual]
```

Get all motor settings in a property value pair dictionary

Returns

Dictionary of properties and values

Implemented in RCCM.TrioStepperZMotor, RCCM.TrioStepperMotor, and RCCM.VirtualMotor.

5.17.3.3 GetPos()

```
abstract double RCCM.Motor.GetPos ( )  [pure virtual]
```

Get current position of actuator

Returns

Current position of actuator

Implemented in RCCM.TrioStepperZMotor, RCCM.TrioStepperMotor, and RCCM.VirtualMotor.

5.17.3.4 GetProperty()

```
virtual double RCCM.Motor.GetProperty (
            string property )  [virtual]
```

Gets a motor property value

Parameters

property	Name of the property

Returns

The value of the specified property

Reimplemented in RCCM.TrioStepperZMotor, and RCCM.TrioStepperMotor.

5.17.3.5 GotoHome()

```
void RCCM.Motor.GotoHome ( )
```

Move to user defined home position

5.17.3.6 Initialize()

```
abstract bool RCCM.Motor.Initialize ( )  [pure virtual]
```

Perform any required initialization

Returns

True if initialization was successful

Implemented in RCCM.TrioStepperZMotor, RCCM.TrioStepperMotor, and RCCM.VirtualMotor.

5.17.3.7 Jog()

```
abstract void RCCM.Motor.Jog (
            bool fwd )  [pure virtual]
```

Begin continuously moving actuator at jog speed

Parameters

fwd	Direction of jog - forward if true

Implemented in RCCM.TrioStepperZMotor, RCCM.VirtualMotor, and RCCM.TrioStepperMotor.

5.17.3.8 JogStop()

```
abstract void RCCM.Motor.JogStop ( )  [pure virtual]
```

Stop jogging

Implemented in RCCM.TrioStepperZMotor, RCCM.VirtualMotor, and RCCM.TrioStepperMotor.

5.17.3.9 MoveRel()

```
abstract double RCCM.Motor.MoveRel (
            double dist )  [pure virtual]
```

Move actuator specified distance from current position

Parameters

dist	Distance to move

Returns

Previous commanded position

Implemented in RCCM.TrioStepperZMotor, RCCM.TrioStepperMotor, and RCCM.VirtualMotor.

5.17.3.10 SetPos()

```
abstract double RCCM.Motor.SetPos (
            double cmd )  [pure virtual]
```

Set new command position of actuator

Parameters

cmd	New goal position to send to actuator

Returns

Goal position that was set, coerced if input was out of travel range

Implemented in RCCM.TrioStepperZMotor, RCCM.TrioStepperMotor, and RCCM.VirtualMotor.

5.17.3.11 SetProperty()

```
virtual bool RCCM.Motor.SetProperty (
            string property,
            double value )  [virtual]
```

Set a motor property

Parameters

property	Property name
value	Property value

Returns

True if property set is successful

Reimplemented in RCCM.TrioStepperZMotor, and RCCM.TrioStepperMotor.

5.17.3.12 Terminate()

```
virtual void RCCM.Motor.Terminate ( )  [virtual]
```

Perform any required action to disconnect from motor

Reimplemented in RCCM.TrioStepperZMotor.

5.17.3.13 WaitForEndOfMove()

```
abstract void RCCM.Motor.WaitForEndOfMove ( )  [pure virtual]
```

Blocking function that runs until current actuator move completes

Implemented in RCCM.TrioStepperZMotor, RCCM.TrioStepperMotor, and RCCM.VirtualMotor.

5.17.3.14 Zero()

```
virtual void RCCM.Motor.Zero ( )  [virtual]
```

Define current actuator position as zero

Reimplemented in RCCM.TrioStepperZMotor, and RCCM.TrioStepperMotor.

5.17.4 Member Data Documentation

5.17.4.1 commandPos

```
double RCCM.Motor.commandPos = 0  [protected]
```

Current goal position of actuator

5.17.4.2 MOTOR_SETTINGS

```
string [] RCCM.Motor.MOTOR_SETTINGS = { "enabled", "microstep per mm", "velocity", "jog speed",
"acceleration", "deceleration", "low position limit", "high position limit", "home", "feedback"
}  [static]
```

List of motor setting names that all motor implementations must include

5.17.4.3 settings

```
Dictionary<string, double> RCCM.Motor.settings  [protected]
```

Maintained list of motion settings and limits

5.17.5 Property Documentation

5.17.5.1 Jogging

```
bool RCCM.Motor.Jogging  [get], [protected set]
```

A flag to indicate if the motor is being jogged

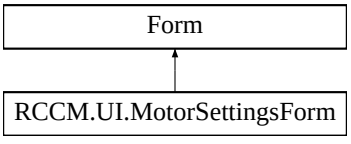
The documentation for this class was generated from the following file:

- RCCM/Motor.cs

5.18 RCCM.UI.MotorSettingsForm Class Reference

Form for adjusting settings of individual actuators

Inheritance diagram for RCCM.UI.MotorSettingsForm:



Public Member Functions

- MotorSettingsForm (RCCMSystem rccm)

  Create form

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

Protected Attributes

- readonly RCCMSystem **rccm**

5.18.1 Detailed Description

Form for adjusting settings of individual actuators

5.18.2 Constructor & Destructor Documentation

5.18.2.1 MotorSettingsForm()

```
RCCM.UI.MotorSettingsForm.MotorSettingsForm (
            RCCMSystem rccm )
```

Create form

Parameters

rccm	Reference to RCCM object

5.18.3 Member Function Documentation

5.18.3.1 Dispose()

```
override void RCCM.UI.MotorSettingsForm.Dispose (
            bool disposing )  [protected]
```

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

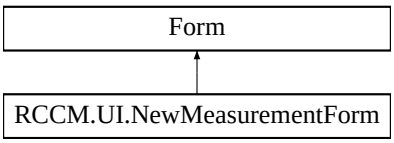
The documentation for this class was generated from the following files:

- RCCM/UI/MotorSettingsForm.cs
- RCCM/UI/MotorSettingsForm.Designer.cs

5.19 RCCM.UI.NewMeasurementForm Class Reference

Form for defining settings for a new MeasurementSequence

Inheritance diagram for RCCM.UI.NewMeasurementForm:



Public Member Functions

- NewMeasurementForm (string defaultName, string camera)
  Open a measurement creation form for the specified camera
- NewMeasurementForm (MeasurementSequence crack)
  Open a form to edit the given crack
- string GetName ()
  Get name defined on form
- Color GetColor ()
  Get color selected with color picker dialog on form
- float GetLineSize ()
  Get line size defined on form
- double GetOrientation ()
  Get crack orientation defined on form
- MeasurementMode GetMode ()
  Get selected measurement mode from form
- string GetCamera ()
  Get name of camera capturing this crack

Static Public Attributes

- static int **ColorInd** = 0
- static Color [ ] **Colors** = { Color.Red, Color.Blue, Color.Green, Color.Purple, Color.Orange, Color.Yellow, Color.Black, Color.Brown, Color.DarkBlue, Color.LimeGreen, Color.PaleVioletRed }

Protected Member Functions

- override void Dispose (bool disposing)
  Clean up any resources being used.

Protected Attributes

- string **camera**

5.19.1 Detailed Description

Form for defining settings for a new MeasurementSequence

5.19.2 Constructor & Destructor Documentation

5.19.2.1 NewMeasurementForm() [1/2]

```
RCCM.UI.NewMeasurementForm.NewMeasurementForm (
            string defaultName,
            string camera )
```

Open a measurement creation form for the specified camera

Parameters

defaultName	Name to show in name field
camera	Name of camera capturing this measurement

5.19.2.2 NewMeasurementForm() [2/2]

```
RCCM.UI.NewMeasurementForm.NewMeasurementForm (
            MeasurementSequence crack )
```

Open a form to edit the given crack

Parameters

crack	Crack to be editted

5.19.3 Member Function Documentation

5.19.3.1 Dispose()

```
override void RCCM.UI.NewMeasurementForm.Dispose (
            bool disposing )  [protected]
```

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

5.19.3.2 GetCamera()

```
string RCCM.UI.NewMeasurementForm.GetCamera ( )
```

Get name of camera capturing this crack

Returns

The name of the camera capturing this MeasurementSequence

5.19.3.3 GetColor()

```
Color RCCM.UI.NewMeasurementForm.GetColor ( )
```

Get color selected with color picker dialog on form

Returns

The selected color

5.19.3.4 GetLineSize()

```
float RCCM.UI.NewMeasurementForm.GetLineSize ( )
```

Get line size defined on form

Returns

The given line size for the MeasurementSequence

5.19.3.5 GetMode()

```
MeasurementMode RCCM.UI.NewMeasurementForm.GetMode ( )
```

Get selected measurement mode from form

Returns

The enum value for the selected measurement mode

5.19.3.6 GetName()

```
string RCCM.UI.NewMeasurementForm.GetName ( )
```

Get name defined on form

Returns

The given name for the MeasurementSequence

5.19.3.7 GetOrientation()

```
double RCCM.UI.NewMeasurementForm.GetOrientation ( )
```

Get crack orientation defined on form

Returns

Orientation angle of crack

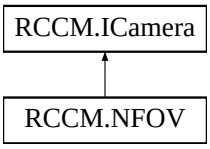
The documentation for this class was generated from the following files:

- RCCM/UI/NewMeasurementForm.cs
- RCCM/UI/NewMeasurementForm.Designer.cs

5.20 RCCM.NFOV Class Reference

Class that handles connecting to and operating NFOV camera (BlackFly

Inheritance diagram for RCCM.NFOV:



Public Member Functions

- NFOV (string name)
  Create a NFOV camera from its serial number and apply the specified calibration
- bool Initialize ()
  Attempt to connect to camera
- void Disconnect ()
  Stop live capture and release camera
- void Start ()
  Start live streaming images from camera
- void Stop ()
  Stop live streaming images
- void ShowPropertiesDlg ()
  Open dialog box for setting camera properties
- void Snap (string filename)
  Save live image to file
- void Record (string aviFileName)
  Begins recording by signalling to grab loop start of recording
- void StopRecord ()
  Stops recording by indicating to grab loop that recording is done
- void SetScale (RCCMSystem rccm, double scale)
  Set image scale and save current height
- bool CheckFOV (RCCMSystem rccm)
  Check if measurement conditions match calibration conditions
- System.Drawing.Bitmap GetLiveImage ()
  Return current live image

Static Public Attributes

- static uint PACKET_SIZE = 4000

  Bytes to send in a single packet. Configured for jumbo packets
- static uint PACKET_DELAY = 6000

  Added delay between packets to prevent dropped frames
- static uint CHANNEL = 0

  Added delay between packets to prevent dropped frames

Protected Attributes

- uint **pixelHeight**
- uint **pixelWidth**
- uint serial

  Serial number of camera
- ManagedGigECamera camera

  Camera object
- ManagedImage rawImage

  Binary image from camera buffer
- bool grabImages

  Flag to direct grab thread to get the live image
- AutoResetEvent grabThreadExited
- BackgroundWorker grabThread

  Background worker for grabbing images from camera
- string videoFileName

  Filename that currently recording video will save to

Properties

- uint PixelHeight `[get, set]`

  Height in pixels of image
- uint PixelWidth `[get, set]`

  Width in pixels of image
- ManagedImage ProcessedImage `[get]`

  Converted image
- bool Connected `[get, protected set]`

  Indicates if camera is connected
- double Scale `[get, protected set]`

  Camera microns / pixel calibration
- double Height `[get]`

  Height in mm of image
- double Width `[get]`

  Height in mm of image
- bool Recording `[get, set]`

  Flag indicating if camera is recording video
- double CalibrationHeight `[get, protected set]`

  Height at which calibration was made

5.20.1 Detailed Description

Class that handles connecting to and operating NFOV camera (BlackFly

5.20.2 Constructor & Destructor Documentation

5.20.2.1 NFOV()

```
RCCM.NFOV.NFOV (
            string name )
```

Create a NFOV camera from its serial number and apply the specified calibration

Parameters

name	Name of camera in settings

5.20.3 Member Function Documentation

5.20.3.1 CheckFOV()

```
bool RCCM.NFOV.CheckFOV (
            RCCMSystem rccm )
```

Check if measurement conditions match calibration conditions

Parameters

rccm	

Returns

True if measurement conditions match calibration

Implements RCCM.ICamera.

5.20.3.2 Disconnect()

```
void RCCM.NFOV.Disconnect ( )
```

Stop live capture and release camera

5.20.3.3 GetLiveImage()

```
System.Drawing.Bitmap RCCM.NFOV.GetLiveImage ( )
```

Return current live image

Returns

Live image as a bitmap

5.20.3.4 Initialize()

```
bool RCCM.NFOV.Initialize ( )
```

Attempt to connect to camera

Returns

True if connection is successful

5.20.3.5 Record()

```
void RCCM.NFOV.Record (
            string aviFileName )
```

Begins recording by signalling to grab loop start of recording

Parameters

aviFileName	Path to .avi file where video will save

Implements RCCM.ICamera.

5.20.3.6 SetScale()

```
void RCCM.NFOV.SetScale (
            RCCMSystem rccm,
            double scale )
```

Set image scale and save current height

Parameters

rccm	
scale	New calibration

Implements RCCM.ICamera.

5.20.3.7 ShowPropertiesDlg()

```
void RCCM.NFOV.ShowPropertiesDlg ( )
```

Open dialog box for setting camera properties

5.20.3.8 Snap()

```
void RCCM.NFOV.Snap (
            string filename )
```

Save live image to file

Parameters

filename	Full path where image will be saved

Implements RCCM.ICamera.

5.20.3.9 Start()

```
void RCCM.NFOV.Start ( )
```

Start live streaming images from camera

Implements RCCM.ICamera.

5.20.3.10 Stop()

```
void RCCM.NFOV.Stop ( )
```

Stop live streaming images

Implements RCCM.ICamera.

5.20.3.11 StopRecord()

```
void RCCM.NFOV.StopRecord ( )
```

Stops recording by indicating to grab loop that recording is done

Implements RCCM.ICamera.

5.20.4 Member Data Documentation

5.20.4.1 camera

```
ManagedGigECamera RCCM.NFOV.camera  [protected]
```

Camera object

5.20.4.2 CHANNEL

```
uint RCCM.NFOV.CHANNEL = 0  [static]
```

Added delay between packets to prevent dropped frames

5.20.4.3 grabImages

```
bool RCCM.NFOV.grabImages  [protected]
```

Flag to direct grab thread to get the live image

5.20.4.4 grabThread

```
BackgroundWorker RCCM.NFOV.grabThread  [protected]
```

Background worker for grabbing images from camera

5.20.4.5 grabThreadExited

```
AutoResetEvent RCCM.NFOV.grabThreadExited  [protected]
```

5.20.4.6 PACKET_DELAY

```
uint RCCM.NFOV.PACKET_DELAY = 6000  [static]
```

Added delay between packets to prevent dropped frames

5.20.4.7 PACKET_SIZE

```
uint RCCM.NFOV.PACKET_SIZE = 4000  [static]
```

Bytes to send in a single packet. Configured for jumbo packets

5.20.4.8 rawImage

```
ManagedImage RCCM.NFOV.rawImage  [protected]
```

Binary image from camera buffer

5.20.4.9 serial

```
uint RCCM.NFOV.serial  [protected]
```

Serial number of camera

5.20.4.10 videoFileName

```
string RCCM.NFOV.videoFileName  [protected]
```

Filename that currently recording video will save to

5.20.5 Property Documentation

5.20.5.1 CalibrationHeight

```
double RCCM.NFOV.CalibrationHeight  [get], [protected set]
```

Height at which calibration was made

5.20.5.2 Connected

```
bool RCCM.NFOV.Connected  [get], [protected set]
```

Indicates if camera is connected

5.20.5.3 Height

```
double RCCM.NFOV.Height  [get]
```

Height in mm of image

5.20.5.4 PixelHeight

```
uint RCCM.NFOV.PixelHeight  [get], [set]
```

Height in pixels of image

5.20.5.5 PixelWidth

```
uint RCCM.NFOV.PixelWidth  [get], [set]
```

Width in pixels of image

5.20.5.6 ProcessedImage

```
ManagedImage RCCM.NFOV.ProcessedImage  [get]
```

Converted image

5.20.5.7 Recording

```
bool RCCM.NFOV.Recording  [get], [set]
```

Flag indicating if camera is recording video

5.20.5.8 Scale

```
double RCCM.NFOV.Scale  [get], [protected set]
```

Camera microns / pixel calibration

5.20.5.9 Width

```
double RCCM.NFOV.Width  [get]
```

Height in mm of image

The documentation for this class was generated from the following file:

- RCCM/NFOV.cs

5.21 RCCM.NFOVLensController Class Reference

Class for operating Gardasoft controller and focusing both optotune liquid lenses

Public Member Functions

- NFOVLensController (int nfov1Serial, int nfov2Serial, double[ ] conversion1, double[ ] conversion2, double[,] calibration1, double[,] calibration2)

  Initialize autofocusing of NFOV lens controllers
- double GetReading (RCCMStage stage)

  Get current input voltage to distanc sensor of specified NFOV
- double GetHeight (RCCMStage stage)

  Get current height of specified NFOV
- double GetFocalPower (RCCMStage stage)

  Get current focal power of specified NFOV lens
- bool ApplyCalibration (double[,] data, RCCMStage stage)

  Applies a calibration to the lens. This scales the focal power based off the distance sensor reading according to a piecewise linear interpolant
- bool SetFocalPower (double power, RCCMStage stage)

  Sends command to specified controller to set a constant focal power value
- double ToHeight1 (double inputVoltage)

  Convert input voltage to distance for NFOV 1
- double ToHeight2 (double inputVoltage)

  Convert input voltage to distance for NFOV 2
- void PauseFocusing (RCCMStage stage)

  Pause autofocus loop for specified stage
- void ResumeFocusing (RCCMStage stage)

  Unpause autofocus loop for specified stage
- void Save ()

  Save calibrations to settings
- void Stop ()

  Stops the focusing thread. Should be called when exiting program

Static Public Member Functions

- static double PwlInterp (double[,] data, double val)

  Helper function for interpolating calibration

Static Public Attributes

- static string GET_VOLTAGE_CMD = "AN3"

  Gardasoft command string for getting current input voltage
- static string GET_STATUS_CMD = "ST"

  Gardasoft command string for getting current status (including output focal power)
- static string SET_FOCALPOWER_CMD = "RS1"

  Gardasoft command string for setting output focal power
- static Regex PARSE_GET = new Regex(@"lg([0-9]+)")

  Regex for parsing get status command response for input voltage
- static Regex PARSE_GETOUTPUT = new Regex(@"AF(\s)+([0-9.-]+)")

Regex for parsing status command for output focal power

- static long UPDATE_PERIOD = (long)Program.Settings.json["distance sensor"]["focus update period"]

  Setting for period between focal power updates
- static double MIN_HEIGHT = (double)Program.Settings.json["distance sensor"]["min height"]

  Minimum height value to send when reading is out of range
- static double MAX_HEIGHT = (double)Program.Settings.json["distance sensor"]["max height"]

  Maximum height value to send when reading is out of range
- static double alpha = (double)Program.Settings.json["distance sensor"]["height reading filter constant"]

  Exponential moving average filter constant applied to distance reading

Protected Attributes

- ControllerManager manager

  Gardasoft controller manager for detecting lens controllers
- bool read

  Flag to indicate if input power should be read
- bool readThread1Paused

  Flag to indicate to background thread if it should autofocus NFOV 1
- bool readThread2Paused

  Flag to indicate to background thread if it should autofocus NFOV 2
- BackgroundWorker bw1

  Background worker for focusing NFOV 1
- BackgroundWorker bw2

  Background worker for focusing NFOV 2
- AutoResetEvent readHeight1ThreadExited

  Event handler for when NFOV 1 background thread exits
- AutoResetEvent readHeight2ThreadExited

  Event handler for when NFOV 2 background thread exits

Properties

- IController NFOV1Controller `[get]`

  Controller interface for NFOV 1
- IController NFOV2Controller `[get]`

  Controller interface for NFOV 2
- double FocusOffset1 `[get, set]`

  Adjustment offset to add to output focal power for NFOV 1
- double FocusOffset2 `[get, set]`

  Adjustment offset to add to output focal power for NFOV 2
- double [ ] conversion1 `[get]`

  Slope, y-intercept of conversion from input voltage to distance (NFOV 1)
- double [ ] conversion2 `[get]`

  Slope, y-intercept of conversion from input voltage to distance (NFOV 2)
- double [,] NFOV1Calibration `[get]`

  Array of input voltage, output focal power pairs to interpolate for NFOV 1
- double [,] NFOV2Calibration `[get]`

  Array of input voltage, output focal power pairs to interpolate for NFOV 2
- double Height1 `[get]`

  Current height reading for NFOV 1
- double Height2 `[get]`

Current height reading for NFOV 2

- TrioStepperZMotor Motor1 [get, set]

  Z motor for NFOV 1

- TrioStepperZMotor Motor2 [get, set]

  Z motor for NFOV 2

5.21.1 Detailed Description

Class for operating Gardasoft controller and focusing both optotune liquid lenses

5.21.2 Constructor & Destructor Documentation

5.21.2.1 NFOVLensController()

```
RCCM.NFOVLensController.NFOVLensController (
            int nfov1Serial,
            int nfov2Serial,
            double [] conversion1,
            double [] conversion2,
            double calibration1[,],
            double calibration2[,] )
```

Initialize autofocusing of NFOV lens controllers

Parameters

nfov1Serial	Serial number of NFOV 1 gardasoft controller
nfov2Serial	Serial number of NFOV 2 gardasoft controller
conversion1	Conversion loaded from settings for NFOV 1
conversion2	Conversion loaded from settings for NFOV 2
calibration1	Lens calibration loaded from settings for NFOV 1
calibration2	Lens calibration loaded from settings for NFOV 2

5.21.3 Member Function Documentation

5.21.3.1 ApplyCalibration()

```
bool RCCM.NFOVLensController.ApplyCalibration (
            double data[,],
            RCCMStage stage )
```

Applies a calibration to the lens. This scales the focal power based off the distance sensor reading according to a piecewise linear interpolant

Parameters

data	2D array of calibration data. Column 1 contains input voltage, column 2 contains output voltage
stage	Parent stage of the lens to indicate which lens to calibrate

Returns

True if calibration was valid

5.21.3.2 GetFocalPower()

```
double RCCM.NFOVLensController.GetFocalPower (
            RCCMStage stage )
```

Get current focal power of specified NFOV lens

Parameters

stage	Enum value of desired lens

Returns

Current focal power output

5.21.3.3 GetHeight()

```
double RCCM.NFOVLensController.GetHeight (
            RCCMStage stage )
```

Get current height of specified NFOV

Parameters

stage	Enum value of desired stage

Returns

Current focal height in user units

5.21.3.4 GetReading()

```
double RCCM.NFOVLensController.GetReading (
            RCCMStage stage )
```

Get current input voltage to distanc sensor of specified NFOV

Parameters

stage	Enum value of desired lens

Returns

Current input voltaget

5.21.3.5 PauseFocusing()

```
void RCCM.NFOVLensController.PauseFocusing (
            RCCMStage stage )
```

Pause autofocus loop for specified stage

Parameters

stage	Enum value corresponding to desired stage

5.21.3.6 PwlInterp()

```
static double RCCM.NFOVLensController.PwlInterp (
            double data[,],
            double val )  [static]
```

Helper function for interpolating calibration

Parameters

data	Array of x, y pairs
val	X value to be interpolated

Returns

Interpolated Y value

5.21.3.7 ResumeFocusing()

```
void RCCM.NFOVLensController.ResumeFocusing (
            RCCMStage stage )
```

Unpause autofocus loop for specified stage

Parameters

stage	Enum value corresponding to desired stage

5.21.3.8 Save()

```
void RCCM.NFOVLensController.Save ( )
```

Save calibrations to settings

5.21.3.9 SetFocalPower()

```
bool RCCM.NFOVLensController.SetFocalPower (
            double power,
            RCCMStage stage )
```

Sends command to specified controller to set a constant focal power value

Parameters

power	Focal power to use, in diopters
stage	Stage indicating which controller to send the command to

Returns

True if command succeded

5.21.3.10 Stop()

```
void RCCM.NFOVLensController.Stop ( )
```

Stops the focusing thread. Should be called when exiting program

5.21.3.11 ToHeight1()

```
double RCCM.NFOVLensController.ToHeight1 (
            double inputVoltage )
```

Convert input voltage to distance for NFOV 1

Parameters

inputVoltage	Input voltage from distance sensor on NFOV 1

Returns

Height corresponding to input

5.21.3.12 ToHeight2()

```
double RCCM.NFOVLensController.ToHeight2 (
            double inputVoltage )
```

Convert input voltage to distance for NFOV 2

Parameters

inputVoltage	Input voltage from distance sensor on NFOV 2

Returns

Height corresponding to input

5.21.4 Member Data Documentation

5.21.4.1 alpha

```
double RCCM.NFOVLensController.alpha = (double)Program.Settings.json["distance sensor"]["height
reading filter constant"]  [static]
```

Exponential moving average filter constant applied to distance reading

5.21.4.2 bw1

```
BackgroundWorker RCCM.NFOVLensController.bw1  [protected]
```

Background worker for focusing NFOV 1

5.21.4.3 bw2

```
BackgroundWorker RCCM.NFOVLensController.bw2  [protected]
```

Background worker for focusing NFOV 2

5.21.4.4 GET_STATUS_CMD

```
string RCCM.NFOVLensController.GET_STATUS_CMD = "ST"  [static]
```

Gardasoft command string for getting current status (including output focal power)

5.21.4.5 GET_VOLTAGE_CMD

```
string RCCM.NFOVLensController.GET_VOLTAGE_CMD = "AN3"  [static]
```

Gardasoft command string for getting current input voltage

5.21.4.6 manager

```
ControllerManager RCCM.NFOVLensController.manager  [protected]
```

Gardasoft controller manager for detecting lens controllers

5.21.4.7 MAX_HEIGHT

```
double RCCM.NFOVLensController.MAX_HEIGHT = (double)Program.Settings.json["distance sensor"]["max
height"]  [static]
```

Maximum height value to send when reading is out of range

5.21.4.8 MIN_HEIGHT

```
double RCCM.NFOVLensController.MIN_HEIGHT = (double)Program.Settings.json["distance sensor"]["min
height"]  [static]
```

Minimum height value to send when reading is out of range

5.21.4.9 PARSE_GET

```
Regex RCCM.NFOVLensController.PARSE_GET = new Regex(@"Ig([0-9]+)")  [static]
```

Regex for parsing get status command response for input voltage

5.21.4.10 PARSE_GETOUTPUT

```
Regex RCCM.NFOVLensController.PARSE_GETOUTPUT = new Regex(@"AF(\s)+([0-9.-]+)")  [static]
```

Regex for parsing status command for output focal power

5.21.4.11 read

```
bool RCCM.NFOVLensController.read  [protected]
```

Flag to indicate if input power should be read

5.21.4.12 readHeight1ThreadExited

```
AutoResetEvent RCCM.NFOVLensController.readHeight1ThreadExited  [protected]
```

Event handler for when NFOV 1 background thread exits

5.21.4.13 readHeight2ThreadExited

```
AutoResetEvent RCCM.NFOVLensController.readHeight2ThreadExited  [protected]
```

Event handler for when NFOV 2 background thread exits

5.21.4.14 readThread1Paused

```
bool RCCM.NFOVLensController.readThread1Paused  [protected]
```

Flag to indicate to background thread if it should autofocus NFOV 1

5.21.4.15 readThread2Paused

```
bool RCCM.NFOVLensController.readThread2Paused  [protected]
```

Flag to indicate to background thread if it should autofocus NFOV 2

5.21.4.16 SET_FOCALPOWER_CMD

```
string RCCM.NFOVLensController.SET_FOCALPOWER_CMD = "RS1"  [static]
```

Gardasoft command string for setting output focal power

5.21.4.17 UPDATE_PERIOD

```
long RCCM.NFOVLensController.UPDATE_PERIOD = (long)Program.Settings.json["distance sensor"]["focus
update period"]  [static]
```

Setting for period between focal power updates

5.21.5 Property Documentation

5.21.5.1 conversion1

```
double [] RCCM.NFOVLensController.conversion1  [get]
```

Slope, y-intercept of conversion from input voltage to distance (NFOV 1)

5.21.5.2 conversion2

```
double [] RCCM.NFOVLensController.conversion2  [get]
```

Slope, y-intercept of conversion from input voltage to distance (NFOV 2)

5.21.5.3 FocusOffset1

```
double RCCM.NFOVLensController.FocusOffset1  [get], [set]
```

Adjustment offset to add to output focal power for NFOV 1

5.21.5.4 FocusOffset2

```
double RCCM.NFOVLensController.FocusOffset2  [get], [set]
```

Adjustment offset to add to output focal power for NFOV 2

5.21.5.5 Height1

```
double RCCM.NFOVLensController.Height1  [get]
```

Current height reading for NFOV 1

5.21.5.6 Height2

```
double RCCM.NFOVLensController.Height2  [get]
```

Current height reading for NFOV 2

5.21.5.7 Motor1

```
TrioStepperZMotor RCCM.NFOVLensController.Motor1  [get], [set]
```

Z motor for NFOV 1

5.21.5.8 Motor2

```
TrioStepperZMotor RCCM.NFOVLensController.Motor2  [get], [set]
```

Z motor for NFOV 2

5.21.5.9 NFOV1Calibration

```
double [,] RCCM.NFOVLensController.NFOV1Calibration  [get]
```

Array of input voltage, output focal power pairs to interpolate for NFOV 1

5.21.5.10 NFOV1Controller

```
IController RCCM.NFOVLensController.NFOV1Controller  [get]
```

Controller interface for NFOV 1

5.21.5.11 NFOV2Calibration

```
double [,] RCCM.NFOVLensController.NFOV2Calibration  [get]
```

Array of input voltage, output focal power pairs to interpolate for NFOV 2

5.21.5.12 NFOV2Controller

```
IController RCCM.NFOVLensController.NFOV2Controller  [get]
```

Controller interface for NFOV 2

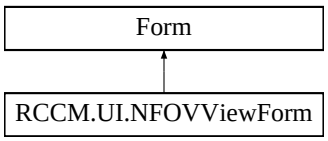
The documentation for this class was generated from the following file:

- RCCM/NFOVLensController.cs

5.22 RCCM.UI.NFOVViewForm Class Reference

Form for displaying NFOV live image and measurement overlay

Inheritance diagram for RCCM.UI.NFOVViewForm:



Public Member Functions

- NFOVViewForm (RCCMSystem rccm, NFOV camera, ObservableCollection< MeasurementSequence > cracks)

  Initialize NFOV display
- void createSegment ()

  Create segment that was being drawn by user with mouse input. This will add a point or segment to the active MeasurementSequence
- void moveDrawnLineEnd (int x, int y, int w, int h)

  Move the end point of the line segment that user is currently drawing
- void createDrawnLine (int x, int y, int w, int h)

  Begin drawing new segment in active MeasurementSequence from user mouse input
- void createPoint (int x, int y, int w, int h)

  Create new point in active measurement sequence at mouse location

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

Protected Attributes

- readonly RCCMSystem rccm

  The RCCM system object. Used for calculating NFOV position
- readonly ObservableCollection< MeasurementSequence > cracks

  List of cracks that are being measured
- readonly NFOV camera

  NFOV camera that is displayed by this window
- RCCMStage stage

  Parent stage of this camera
- PointF drawnLineStart

  Point where line user is drawing begins
- PointF drawnLineEnd

  Point where line user is drawing ends
- Timer nfovRepaintTimer

  Timer for calling NFOV display repaint

Properties

- int ActiveIndex `[get, set]`

  Currently selected crack index. The selected crack will be edited by other controls
- int ActivePoint `[get, set]`

  Currently selected point in ListPoints. This point is indicated in the image display
- float displayScale `[get]`

  Ratio of picture box size to actual NFOV image dimensions
- bool Drawing `[get]`

  Indicates whether or not user is drawing a line with mouse

5.22.1 Detailed Description

Form for displaying NFOV live image and measurement overlay

5.22.2 Constructor & Destructor Documentation

5.22.2.1 NFOVViewForm()

```
RCCM.UI.NFOVViewForm.NFOVViewForm (
            RCCMSystem rccm,
            NFOV camera,
            ObservableCollection< MeasurementSequence > cracks )
```

Initialize NFOV display

Parameters

rccm	RCCMSystem object, needed for getting location and zoom status
camera	NFOV camera to display
cracks	List of cracks to display

5.22.3 Member Function Documentation

5.22.3.1 createDrawnLine()

```
void RCCM.UI.NFOVViewForm.createDrawnLine (
            int x,
            int y,
            int w,
            int h )
```

Begin drawing new segment in active MeasurementSequence from user mouse input

Parameters

x	Mouse x location in pixels
y	Mouse y location in pixels
w	Canvas width in pixels
h	Canvas height in pixels

5.22.3.2 createPoint()

```
void RCCM.UI.NFOVViewForm.createPoint (
            int x,
            int y,
            int w,
            int h )
```

Create new point in active measurement sequence at mouse location

Parameters

x	Mouse x location in pixels
y	Mouse y location in pixels
w	Canvas width in pixels
h	Canvas height in pixels

5.22.3.3 createSegment()

```
void RCCM.UI.NFOVViewForm.createSegment ( )
```

Create segment that was being drawn by user with mouse input. This will add a point or segment to the active MeasurementSequence

5.22.3.4 Dispose()

```
override void RCCM.UI.NFOVViewForm.Dispose (
            bool disposing )  [protected]
```

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

5.22.3.5 moveDrawnLineEnd()

```
void RCCM.UI.NFOVViewForm.moveDrawnLineEnd (
            int x,
            int y,
            int w,
            int h )
```

Move the end point of the line segment that user is currently drawing

Parameters

x	Mouse x location in pixels
y	Mouse y location in pixels
w	Canvas width in pixels
h	Canvas height in pixels

5.22.4 Member Data Documentation

5.22.4.1 camera

```
readonly NFOV RCCM.UI.NFOVViewForm.camera  [protected]
```

NFOV camera that is displayed by this window

5.22.4.2 cracks

```
readonly ObservableCollection<MeasurementSequence> RCCM.UI.NFOVViewForm.cracks  [protected]
```

List of cracks that are being measured

5.22.4.3 drawnLineEnd

```
PointF RCCM.UI.NFOVViewForm.drawnLineEnd  [protected]
```

Point where line user is drawing ends

5.22.4.4 drawnLineStart

```
PointF RCCM.UI.NFOVViewForm.drawnLineStart  [protected]
```

Point where line user is drawing begins

5.22.4.5 nfovRepaintTimer

```
Timer RCCM.UI.NFOVViewForm.nfovRepaintTimer  [protected]
```

Timer for calling NFOV display repaint

5.22.4.6 rccm

```
readonly RCCMSystem RCCM.UI.NFOVViewForm.rccm  [protected]
```

The RCCM system object. Used for calculating NFOV position

5.22.4.7 stage

```
RCCMStage RCCM.UI.NFOVViewForm.stage  [protected]
```

Parent stage of this camera

5.22.5 Property Documentation

5.22.5.1 ActiveIndex

```
int RCCM.UI.NFOVViewForm.ActiveIndex  [get], [set], [protected]
```

Currently selected crack index. The selected crack will be edited by other controls

5.22.5.2 ActivePoint

```
int RCCM.UI.NFOVViewForm.ActivePoint  [get], [set], [protected]
```

Currently selected point in ListPoints. This point is indicated in the image display

5.22.5.3 displayScale

```
float RCCM.UI.NFOVViewForm.displayScale  [get], [protected]
```

Ratio of picture box size to actual NFOV image dimensions

5.22.5.4 Drawing

```
bool RCCM.UI.NFOVViewForm.Drawing  [get], [protected]
```

Indicates whether or not user is drawing a line with mouse

The documentation for this class was generated from the following files:

- RCCM/UI/NFOVViewForm.cs
- RCCM/UI/NFOVViewForm.Designer.cs

5.23 RCCM.PanelView Class Reference

Object used to draw panel graphic to show location of stages

Public Member Functions

- PanelView (RCCMSystem rccm)

  Initialize panel view
- void Paint (Graphics g)

  Paint panel, stages, and position crosshair graphics
- void SetTransform (Graphics g)

  Create the transform matrix to map pixel coordinates to the global coordinate system

Protected Attributes

- readonly RCCMSystem rccm

  Reference to RCCMSystem object
- Pen pen

  Pen for drawing lines
- Brush panelBrush

  Brush defining style for drawing panel rectangle
- Brush coarseBrush

  Brush defining style for drawing coarse stage rectangle
- Brush fineBrush

  Brush defining style for drawing fine stage rectangles
- RectangleF panel

  Rectangle defining dimensions of panel
- RectangleF coarse

  Rectangle defining dimensions of coarse stage travel region
- RectangleF fine1

  Rectangle defining dimensions of fine 1 stage travel region
- RectangleF fine2

  Rectangle defining dimensions of fine 1 stage travel region
- SizeF fine1Offset

  Position offset to fine 1 rectangle top left corner
- SizeF fine2Offset

  Position offset to fine 2 rectangle top left corner
- Matrix transform

Graphics transform mapping pixels to global coordinate system

- float coarseXPos

  Coarse X actuator position
- float coarseYPos

  Coarse Y actuator position
- float fine1XPos

  Fine 1 X actuator position
- float fine1YPos

  Fine 1 Y actuator position
- float fine2XPos

  Fine 2 X actuator position
- float fine2YPos

  Fine 2 Y actuator position

5.23.1 Detailed Description

Object used to draw panel graphic to show location of stages

5.23.2 Constructor & Destructor Documentation

5.23.2.1 PanelView()

```
RCCM.PanelView.PanelView (
            RCCMSystem rccm )
```

Initialize panel view

Parameters

rccm	Reference to RCCM object

5.23.3 Member Function Documentation

5.23.3.1 Paint()

```
void RCCM.PanelView.Paint (
            Graphics g )
```

Paint panel, stages, and position crosshair graphics

Parameters

g	The graphics object representing the control on which to draw
rccm	Handle to RCCMSystem object. Will be used to get positions

5.23.3.2 SetTransform()

```
void RCCM.PanelView.SetTransform (
            Graphics g )
```

Create the transform matrix to map pixel coordinates to the global coordinate system

Parameters

g	The graphics object representing the control on which to draw

5.23.4 Member Data Documentation

5.23.4.1 coarse

```
RectangleF RCCM.PanelView.coarse  [protected]
```

Rectangle defining dimensions of coarse stage travel region

5.23.4.2 coarseBrush

```
Brush RCCM.PanelView.coarseBrush  [protected]
```

Brush defining style for drawing coarse stage rectangle

5.23.4.3 coarseXPos

```
float RCCM.PanelView.coarseXPos  [protected]
```

Coarse X actuator position

5.23.4.4 coarseYPos

```
float RCCM.PanelView.coarseYPos  [protected]
```

Coarse Y actuator position

5.23.4.5 fine1

```
RectangleF RCCM.PanelView.fine1  [protected]
```

Rectangle defining dimensions of fine 1 stage travel region

5.23.4.6 fine1Offset

```
SizeF RCCM.PanelView.fine1Offset  [protected]
```

Position offset to fine 1 rectangle top left corner

5.23.4.7 fine1XPos

```
float RCCM.PanelView.fine1XPos  [protected]
```

Fine 1 X actuator position

5.23.4.8 fine1YPos

```
float RCCM.PanelView.fine1YPos  [protected]
```

Fine 1 Y actuator position

5.23.4.9 fine2

```
RectangleF RCCM.PanelView.fine2  [protected]
```

Rectangle defining dimensions of fine 1 stage travel region

5.23.4.10 fine2Offset

```
SizeF RCCM.PanelView.fine2Offset  [protected]
```

Position offset to fine 2 rectangle top left corner

5.23.4.11 fine2XPos

```
float RCCM.PanelView.fine2XPos  [protected]
```

Fine 2 X actuator position

5.23.4.12 fine2YPos

```
float RCCM.PanelView.fine2YPos  [protected]
```

Fine 2 Y actuator position

5.23.4.13 fineBrush

```
Brush RCCM.PanelView.fineBrush  [protected]
```

Brush defining style for drawing fine stage rectangles

5.23.4.14 panel

```
RectangleF RCCM.PanelView.panel  [protected]
```

Rectangle defining dimensions of panel

5.23.4.15 panelBrush

```
Brush RCCM.PanelView.panelBrush  [protected]
```

Brush defining style for drawing panel rectangle

5.23.4.16 pen

```
Pen RCCM.PanelView.pen  [protected]
```

Pen for drawing lines

5.23.4.17 rccm

```
readonly RCCMSystem RCCM.PanelView.rccm  [protected]
```

Reference to RCCMSystem object

5.23.4.18 transform

```
Matrix RCCM.PanelView.transform  [protected]
```

Graphics transform mapping pixels to global coordinate system

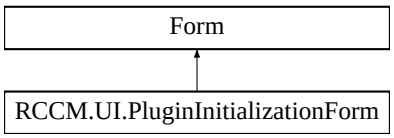
The documentation for this class was generated from the following file:

- RCCM/PanelView.cs

5.24 RCCM.UI.PluginInitializationForm Class Reference

Form for starting plugin and entering inputs

Inheritance diagram for RCCM.UI.PluginInitializationForm:



Public Member Functions

- PluginInitializationForm (RCCMSystem rccm, IRCCMPlugin plugin)
  Open form for given plugin

Protected Member Functions

- override void Dispose (bool disposing)
  Clean up any resources being used.

Protected Attributes

- readonly RCCMSystem **rccm**
- readonly IRCCMPlugin **plugin**
- IRCCMPluginActor **actor**
- TableLayoutPanel **tableLayoutPanelParams**
- Dictionary< string, TextBox > **parameterControls**

5.24.1 Detailed Description

Form for starting plugin and entering inputs

5.24.2 Constructor & Destructor Documentation

5.24.2.1 PluginInitializationForm()

```
RCCM.UI.PluginInitializationForm.PluginInitializationForm (
            RCCMSystem rccm,
            IRCCMPlugin plugin )
```

Open form for given plugin

Parameters

rccm	Reference to RCCM object
plugin	Plugin to be started from this form

5.24.3 Member Function Documentation

5.24.3.1 Dispose()

```
override void RCCM.UI.PluginInitializationForm.Dispose (
            bool disposing )  [protected]
```

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

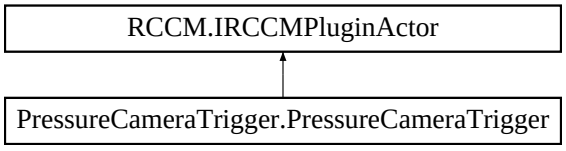
The documentation for this class was generated from the following files:

- RCCM/UI/PluginInitializationForm.cs
- RCCM/UI/PluginInitializationForm.Designer.cs

5.25 PressureCameraTrigger.PressureCameraTrigger Class Reference

Captures images when triggger pressure is reached

Inheritance diagram for PressureCameraTrigger.PressureCameraTrigger:



Public Member Functions

- PressureCameraTrigger (RCCMSystem rccm, Dictionary< string, string > parameters)
  Create plugin actor from parameter strings
- void Run ()
  Continuously read pressure and capture image when threshold is passed
- void Stop ()
  This function should cause the plugin to stop as soon as possible

Protected Attributes

- readonly ICamera **camera**
- string **path**
- double **triggerPressure**
- bool **ascending**
- readonly RCCMSystem **rccm**

Properties

- bool **Running** `[get, protected set]`

5.25.1 Detailed Description

Captures images when triggger pressure is reached

5.25.2 Constructor & Destructor Documentation

5.25.2.1 PressureCameraTrigger()

```
PressureCameraTrigger.PressureCameraTrigger.PressureCameraTrigger (
            RCCMSystem rccm,
            Dictionary< string, string > parameters )
```

Create plugin actor from parameter strings

Parameters

rccm	Reference to RCCM object
parameters	Map of test parameters to values

5.25.3 Member Function Documentation

5.25.3.1 Run()

```
void PressureCameraTrigger.PressureCameraTrigger.Run ( )
```

Continuously read pressure and capture image when threshold is passed

Implements RCCM.IRCCMPluginActor.

5.25.3.2 Stop()

```
void PressureCameraTrigger.PressureCameraTrigger.Stop ( )
```

This function should cause the plugin to stop as soon as possible

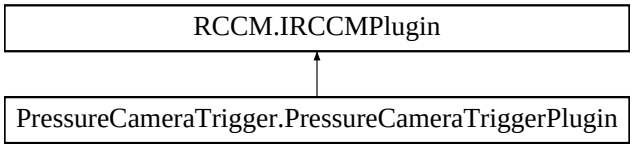
Implements RCCM.IRCCMPluginActor.

The documentation for this class was generated from the following file:

- RCCM/Plugins/PressureCameraTrigger/PressureCameraTrigger/PressureCameraTrigger.cs

5.26 PressureCameraTrigger.PressureCameraTriggerPlugin Class Reference

Inheritance diagram for PressureCameraTrigger.PressureCameraTriggerPlugin:



Public Member Functions

- IRCCMPluginActor Instance (RCCMSystem rccm, Dictionary< string, string > parameters)
  Create plugin with given test parameters

Static Public Attributes

- static double PERIOD = 50
  Time period of a single loop iteration

Properties

- string Name [get]
  Publicly visible plugin name
- string [ ] Params [get]
  User entered test inputs

5.26.1 Member Function Documentation

5.26.1.1 Instance()

```
IRCCMPluginActor PressureCameraTrigger.PressureCameraTriggerPlugin.Instance (
            RCCMSystem rccm,
            Dictionary< string, string > parameters )
```

Create plugin with given test parameters

Parameters

rccm	
parameters	User entered test parameters

Returns

Implements RCCM.IRCCMPlugin.

5.26.2 Member Data Documentation

5.26.2.1 PERIOD

```
double PressureCameraTrigger.PressureCameraTriggerPlugin.PERIOD = 50  [static]
```

Time period of a single loop iteration

5.26.3 Property Documentation

5.26.3.1 Name

```
string PressureCameraTrigger.PressureCameraTriggerPlugin.Name  [get]
```

Publicly visible plugin name

5.26.3.2 Params

```
string [] PressureCameraTrigger.PressureCameraTriggerPlugin.Params  [get]
```

User entered test inputs

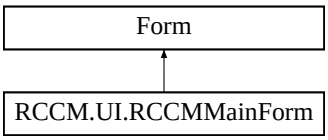
The documentation for this class was generated from the following file:

- RCCM/Plugins/PressureCameraTrigger/PressureCameraTrigger/PressureCameraTriggerPlugin.cs

5.27 RCCM.UI.RCCMMainForm Class Reference

The main window of the program from which all hardware initialization and termination is directed

Inheritance diagram for RCCM.UI.RCCMMainForm:



Classes

- class **NativeMethods**

  Helper class fo preventing OS from putting computer in sleep mode

Public Member Functions

- RCCMMainForm (ICollection< IRCCMPlugin > plugins)

  Create the main form and initialize all hardware
- void applyUISettings ()

  Set increment of position controls to actuator step value

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

Protected Attributes

- RCCMSystem rccm

  The main object representing the RCCM containing references to all hardware
- bool nfov1Open

  Flag for indicating if NFOV1 form is open
- bool nfov2Open

  Flag for indicating if NFOV2 form is open
- bool wfov1Open

  Flag for indicating if WFOV1 form is open
- bool wfov2Open

  Flag for indicating if WFOV2 form is open
- Timer panelRepaintTimer

  Timer for replotting certain panel graphic
- ObservableCollection< MeasurementSequence > cracks

  List of measurement objects. Observable so controls can update when other forms update them
- TestResults test

  Object for managing test result plotting
- PanelView view

  Object for managing panel graphic drawing
- ComponentResourceManager resources

  Resource manager for loading ActiveX object for Trio controller
- AxTrioPCLib.AxTrioPC triopc

  ActiveX control for trio controller
- uint fPreviousExecutionState

  Holds sleep setting of computer before starting program so it can be reverted on exit

5.27.1 Detailed Description

The main window of the program from which all hardware initialization and termination is directed

5.27.2 Constructor & Destructor Documentation

5.27.2.1 RCCMMainForm()

```
RCCM.UI.RCCMMainForm.RCCMMainForm (
            ICollection< IRCCMPlugin > plugins )
```

Create the main form and initialize all hardware

Parameters

plugins	List of plugins that were found on startup

5.27.3 Member Function Documentation

5.27.3.1 applyUISettings()

```
void RCCM.UI.RCCMMainForm.applyUISettings ( )
```

Set increment of position controls to actuator step value

5.27.3.2 Dispose()

```
override void RCCM.UI.RCCMMainForm.Dispose (
            bool disposing )  [protected]
```

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

5.27.4 Member Data Documentation

5.27.4.1 cracks

```
ObservableCollection<MeasurementSequence> RCCM.UI.RCCMMainForm.cracks  [protected]
```

List of measurement objects. Observable so controls can update when other forms update them

5.27.4.2 fPreviousExecutionState

```
uint RCCM.UI.RCCMMainForm.fPreviousExecutionState  [protected]
```

Holds sleep setting of computer before starting program so it can be reverted on exit

5.27.4.3 nfov1Open

```
bool RCCM.UI.RCCMMainForm.nfov1Open  [protected]
```

Flag for indicating if NFOV1 form is open

5.27.4.4 nfov2Open

```
bool RCCM.UI.RCCMMainForm.nfov2Open  [protected]
```

Flag for indicating if NFOV2 form is open

5.27.4.5 panelRepaintTimer

```
Timer RCCM.UI.RCCMMainForm.panelRepaintTimer  [protected]
```

Timer for replotting certain panel graphic

5.27.4.6 rccm

```
RCCMSystem RCCM.UI.RCCMMainForm.rccm  [protected]
```

The main object representing the RCCM containing references to all hardware

5.27.4.7 resources

```
ComponentResourceManager RCCM.UI.RCCMMainForm.resources  [protected]
```

Resource manager for loading ActiveX object for Trio controller

5.27.4.8 test

```
TestResults RCCM.UI.RCCMMainForm.test  [protected]
```

Object for managing test result plotting

5.27.4.9 triopc

```
AxTrioPCLib.AxTrioPC RCCM.UI.RCCMMainForm.triopc  [protected]
```

ActiveX control for trio controller

5.27.4.10 view

```
PanelView RCCM.UI.RCCMMainForm.view  [protected]
```

Object for managing panel graphic drawing

5.27.4.11 wfov1Open

```
bool RCCM.UI.RCCMMainForm.wfov1Open  [protected]
```

Flag for indicating if WFOV1 form is open

5.27.4.12 wfov2Open

```
bool RCCM.UI.RCCMMainForm.wfov2Open  [protected]
```

Flag for indicating if WFOV2 form is open

The documentation for this class was generated from the following files:

- RCCM/UI/RCCMMainForm.cs
- RCCM/UI/RCCMMainForm.Designer.cs

5.28 RCCM.RCCMPluginLoader Class Reference

Plugin loading framework from https://code.msdn.microsoft.com/windowsdesktop/↩Creating-a-simple-plugin-b6174b62

Static Public Member Functions

- static ICollection< IRCCMPlugin > LoadPlugins (string path)

  Load plugin dlls from specified path

5.28.1 Detailed Description

Plugin loading framework from https://code.msdn.microsoft.com/windowsdesktop/Creating-a-simple-plugin-b6174b62

5.28.2 Member Function Documentation

5.28.2.1 LoadPlugins()

```
static ICollection<IRCCMPlugin> RCCM.RCCMPluginLoader.LoadPlugins (
            string path )  [static]
```

Load plugin dlls from specified path

Parameters

path	Path where program should search for plugin dlls

Returns

List of plugin interfaces

The documentation for this class was generated from the following file:

- RCCM/RCCMPluginLoader.cs

5.29 RCCM.RCCMSystem Class Reference

Object representing all the hardware and definitions for the RCCM

Public Member Functions

- RCCMSystem (AxTrioPC axTrioPC)
  Create RCCM object and initialize all hardware
- double GetPanelDistance (double panelX, double panelY)
  Calculate expected distance to panel from distance sensor at a given point
- PointF GetNFOVLocation (RCCMStage stage, CoordinateSystem sys)
  Get position of NFOV camera in specified coordinate system
- PointF GetWFOVLocation (RCCMStage stage, CoordinateSystem sys)
  Get position of WFOV camera in specified coordinate system
- PointF FineVectorToGlobalVector (double x, double y)
  Convert position relative to rotation plate to global coordinate system vector
- PointF GlobalVectorToPanelVector (double x, double y)
  Convert position in global coordinate system to panel coordinates
- void ApplyMotorSettings ()
  Apply all defined settings for all axes
- void SaveMotorSettings ()
  Save motor settings to Settings object
- async Task Stop ()
  Perform any action necessary to disconnect from hardware

Static Public Attributes

- static string [ ] AXES = new string[8] { "coarse X", "coarse Y", "fine 1 X", "fine 1 Y", "fine 1 Z", "fine 2 X", "fine 2 Y", "fine 2 Z" }

  String keys used to represent actuators in settings and Motor dictionary

Protected Member Functions

- void setFineStageRotationMatrix ()

  When FineStageAngle is updated, this function is called to update rotation matrix
- void setPanelRotationMatrix ()

  When PanelAngle is updated, this function is called to update rotation matrix

Protected Attributes

- double [,] **fineStageRotation**
- double [,] **panelRotation**

Properties

- NFOV NFOV1 `[get]`

  Narrow field of view camera on the first set of fine axes
- NFOV NFOV2 `[get]`

  Narrow field of view camera on the second set of fine axes
- WFOV WFOV1 `[get]`

  Wide field of view camera on the first set of fine axes
- WFOV WFOV2 `[get]`

  Wide field of view camera on the second set of fine axes
- NFOVLensController LensController `[get]`

  Object for autofocusing NFOV lenses based on distance sensor input
- Dictionary< string, Motor > motors `[get]`

  Dictionary mapping string keys to motor objects
- ICycleCounter Counter `[get]`

  Interface to a real or virtual cycle counter / pressure input
- double FineStageAngle `[get, set]`

  Angle of fine stage beam on rotation plate. Changing this value also updates settings
- double PanelAngle `[get, set]`

  Angle of panel with respect to coarse axes. Changing this value also updates settings
- double PanelOffsetX `[get, set]`

  Axial offset of panel from corner of coarse axis travel region. Changing this value also updates settings
- double PanelOffsetY `[get, set]`

  Hoop offset of panel from corner of coarse axis travel region. Changing this value also updates settings
- double PanelWidth `[get, set]`

  Axial dimension of panel. Changing this value also updates settings
- double PanelHeight `[get, set]`

  Hoop dimension of panel. Changing this value also updates settings
- double PanelRadius `[get, set]`

  Radius of curvature of panel. Changing this value also updates settings
- double NFOV1X `[get, set]`

Offset of NFOV1 from center of rotation plate along beam. Changing this value also updates settings

- double NFOV1Y [get, set]

  Offset of NFOV1 from center of rotation plate perpindicular to beam. Changing this value also updates settings
- double NFOV2X [get, set]

  Offset of NFOV2 from center of rotation plate along beam. Changing this value also updates settings
- double NFOV2Y [get, set]

  Offset of NFOV2 from center of rotation plate perpindicular to beam. Changing this value also updates settings
- double WFOV1X [get, set]

  Offset of WFOV1 from center of rotation plate along beam. Changing this value also updates settings
- double WFOV1Y [get, set]

  Offset of WFOV1 from center of rotation plate perpindicular to beam. Changing this value also updates settings
- double WFOV2X [get, set]

  Offset of WFOV2 from center of rotation plate along beam. Changing this value also updates settings
- double WFOV2Y [get, set]

  Offset of WFOV2 from center of rotation plate perpindicular to beam. Changing this value also updates settings
- TrioController triopc [get]

  Helper object for sending commands to trio controller

5.29.1 Detailed Description

Object representing all the hardware and definitions for the RCCM

5.29.2 Constructor & Destructor Documentation

5.29.2.1 RCCMSystem()

```
RCCM.RCCMSystem.RCCMSystem (
            AxTrioPC axTrioPC )
```

Create RCCM object and initialize all hardware

Parameters

axTrioPC	ActiveX control for communicating with Trio controller

5.29.3 Member Function Documentation

5.29.3.1 ApplyMotorSettings()

```
void RCCM.RCCMSystem.ApplyMotorSettings ( )
```

Apply all defined settings for all axes

5.29.3.2 FineVectorToGlobalVector()

```
PointF RCCM.RCCMSystem.FineVectorToGlobalVector (
            double x,
            double y )
```

Convert position relative to rotation plate to global coordinate system vector

Parameters

x	Position along fine stage beam
y	Position perpindicular to fine stage beam

Returns

PointF with X and Y coordinates in global frame

5.29.3.3 GetNFOVLocation()

```
PointF RCCM.RCCMSystem.GetNFOVLocation (
            RCCMStage stage,
            CoordinateSystem sys )
```

Get position of NFOV camera in specified coordinate system

Parameters

stage	Enum value corresponding to desired camera
sys	Enum value corresponding to desired coordinate system

Returns

PointF with X and Y coordinates of camera

5.29.3.4 GetPanelDistance()

```
double RCCM.RCCMSystem.GetPanelDistance (
            double panelX,
            double panelY )
```

Calculate expected distance to panel from distance sensor at a given point

Parameters

panelX	X coordinate on panel
panelY	Y coordinate on panel

Returns

Distance to panel - may be negative

5.29.3.5 GetWFOVLocation()

```
PointF RCCM.RCCMSystem.GetWFOVLocation (
            RCCMStage stage,
            CoordinateSystem sys )
```

Get position of WFOV camera in specified coordinate system

Parameters

stage	Enum value corresponding to desired camera
sys	Enum value corresponding to desired coordinate system

Returns

PointF with X and Y coordinates of camera

5.29.3.6 GlobalVectorToPanelVector()

```
PointF RCCM.RCCMSystem.GlobalVectorToPanelVector (
            double x,
            double y )
```

Convert position in global coordinate system to panel coordinates

Parameters

x	Position along coarse axial direction
y	Position along coarse hoop direction

Returns

PointF with X and Y coordinates in panel frame

5.29.3.7 SaveMotorSettings()

```
void RCCM.RCCMSystem.SaveMotorSettings ( )
```

Save motor settings to Settings object

5.29.3.8 setFineStageRotationMatrix()

```
void RCCM.RCCMSystem.setFineStageRotationMatrix ( )  [protected]
```

When FineStageAngle is updated, this function is called to update rotation matrix

5.29.3.9 setPanelRotationMatrix()

```
void RCCM.RCCMSystem.setPanelRotationMatrix ( )  [protected]
```

When PanelAngle is updated, this function is called to update rotation matrix

5.29.3.10 Stop()

```
async Task RCCM.RCCMSystem.Stop ( )
```

Perform any action necessary to disconnect from hardware

Returns

Task completion

5.29.4 Member Data Documentation

5.29.4.1 AXES

```
string [] RCCM.RCCMSystem.AXES = new string[8] { "coarse X", "coarse Y", "fine 1 X", "fine 1
Y", "fine 1 Z", "fine 2 X", "fine 2 Y", "fine 2 Z" }  [static]
```

String keys used to represent actuators in settings and Motor dictionary

5.29.5 Property Documentation

5.29.5.1 Counter

```
ICycleCounter RCCM.RCCMSystem.Counter  [get]
```

Interface to a real or virtual cycle counter / pressure input

5.29.5.2 FineStageAngle

```
double RCCM.RCCMSystem.FineStageAngle  [get], [set]
```

Angle of fine stage beam on rotation plate. Changing this value also updates settings

5.29.5.3 LensController

```
NFOVLensController RCCM.RCCMSystem.LensController  [get]
```

Object for autofocusing NFOV lenses based on distance sensor input

5.29.5.4 motors

```
Dictionary<string, Motor> RCCM.RCCMSystem.motors  [get]
```

Dictionary mapping string keys to motor objects

5.29.5.5 NFOV1

```
NFOV RCCM.RCCMSystem.NFOV1  [get]
```

Narrow field of view camera on the first set of fine axes

5.29.5.6 NFOV1X

```
double RCCM.RCCMSystem.NFOV1X  [get], [set]
```

Offset of NFOV1 from center of rotation plate along beam. Changing this value also updates settings

5.29.5.7 NFOV1Y

```
double RCCM.RCCMSystem.NFOV1Y  [get], [set]
```

Offset of NFOV1 from center of rotation plate perpindicular to beam. Changing this value also updates settings

5.29.5.8 NFOV2

```
NFOV RCCM.RCCMSystem.NFOV2  [get]
```

Narrow field of view camera on the second set of fine axes

5.29.5.9 NFOV2X

```
double RCCM.RCCMSystem.NFOV2X  [get], [set]
```

Offset of NFOV2 from center of rotation plate along beam. Changing this value also updates settings

5.29.5.10 NFOV2Y

```
double RCCM.RCCMSystem.NFOV2Y  [get], [set]
```

Offset of NFOV2 from center of rotation plate perpindicular to beam. Changing this value also updates settings

5.29.5.11 PanelAngle

```
double RCCM.RCCMSystem.PanelAngle  [get], [set]
```

Angle of panel with respect to coarse axes. Changing this value also updates settings

5.29.5.12 PanelHeight

```
double RCCM.RCCMSystem.PanelHeight  [get], [set]
```

Hoop dimension of panel. Changing this value also updates settings

5.29.5.13 PanelOffsetX

```
double RCCM.RCCMSystem.PanelOffsetX  [get], [set]
```

Axial offset of panel from corner of coarse axis travel region. Changing this value also updates settings

5.29.5.14 PanelOffsetY

```
double RCCM.RCCMSystem.PanelOffsetY  [get], [set]
```

Hoop offset of panel from corner of coarse axis travel region. Changing this value also updates settings

5.29.5.15 PanelRadius

```
double RCCM.RCCMSystem.PanelRadius  [get], [set]
```

Radius of curvature of panel. Changing this value also updates settings

5.29.5.16 PanelWidth

```
double RCCM.RCCMSystem.PanelWidth  [get], [set]
```

Axial dimension of panel. Changing this value also updates settings

5.29.5.17 triopc

```
TrioController RCCM.RCCMSystem.triopc  [get]
```

Helper object for sending commands to trio controller

5.29.5.18 WFOV1

```
WFOV RCCM.RCCMSystem.WFOV1  [get]
```

Wide field of view camera on the first set of fine axes

5.29.5.19 WFOV1X

```
double RCCM.RCCMSystem.WFOV1X  [get], [set]
```

Offset of WFOV1 from center of rotation plate along beam. Changing this value also updates settings

5.29.5.20 WFOV1Y

```
double RCCM.RCCMSystem.WFOV1Y  [get], [set]
```

Offset of WFOV1 from center of rotation plate perpindicular to beam. Changing this value also updates settings

5.29.5.21 WFOV2

```
WFOV RCCM.RCCMSystem.WFOV2  [get]
```

Wide field of view camera on the second set of fine axes

5.29.5.22 WFOV2X

```
double RCCM.RCCMSystem.WFOV2X  [get], [set]
```

Offset of WFOV2 from center of rotation plate along beam. Changing this value also updates settings

5.29.5.23 WFOV2Y

```
double RCCM.RCCMSystem.WFOV2Y  [get], [set]
```

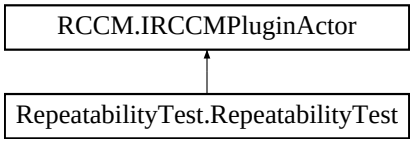
Offset of WFOV2 from center of rotation plate perpindicular to beam. Changing this value also updates settings

The documentation for this class was generated from the following file:

- RCCM/RCCMSystem.cs

5.30 RepeatabilityTest.RepeatabilityTest Class Reference

Inheritance diagram for RepeatabilityTest.RepeatabilityTest:



Public Member Functions

- **RepeatabilityTest** (RCCMSystem rccm, Dictionary< string, string > parameters)
- void Run ()

  Run the function with the inputs already passed to the plugin
- void Stop ()

  This function should cause the plugin to stop as soon as possible

Protected Attributes

- readonly ICamera **camera**
- readonly Motor **motor**
- string **path**
- int **repetitions**
- double **distance**
- readonly RCCMSystem **rccm**
- Thread **testThread**

Properties

- bool **Running** `[get, protected set]`

5.30.1 Member Function Documentation

5.30.1.1 Run()

```
void RepeatabilityTest.RepeatabilityTest.Run ( )
```

Run the function with the inputs already passed to the plugin

Implements RCCM.IRCCMPluginActor.

5.30.1.2 Stop()

```
void RepeatabilityTest.RepeatabilityTest.Stop ( )
```

This function should cause the plugin to stop as soon as possible

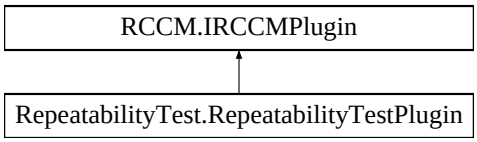
Implements RCCM.IRCCMPluginActor.

The documentation for this class was generated from the following file:

- RCCM/Plugins/RepeatabilityTest/RepeatabilityTest/RepeatabilityTest.cs

5.31 RepeatabilityTest.RepeatabilityTestPlugin Class Reference

Inheritance diagram for RepeatabilityTest.RepeatabilityTestPlugin:



Public Member Functions

- IRCCMPluginActor Instance (RCCMSystem rccm, Dictionary< string, string > parameters)
  Create an instance of the plugin

Properties

- string **Name** `[get]`
- string [ ] **Params** `[get]`

5.31.1 Member Function Documentation

5.31.1.1 Instance()

```
IRCCMPluginActor RepeatabilityTest.RepeatabilityTestPlugin.Instance (
            RCCMSystem rccm,
            Dictionary< string, string > parameters )
```

Create an instance of the plugin

Parameters

rccm	Reference to the RCCM object
parameters	User inputs to the plugin

Returns

An instance of the runnable plugin interface

Implements RCCM.IRCCMPlugin.

The documentation for this class was generated from the following file:

- RCCM/Plugins/RepeatabilityTest/RepeatabilityTest/RepeatabilityTestPlugin.cs

5.32 RCCM.Settings Class Reference

Object representing settings json file

Public Member Functions

- Settings (string filename)
  Create settings object from file
- bool save ()
  Save settings to original file they were read from
  Returns
  True if settings saved successfully, false otherwise
- bool save (string filename)
  Save settings to a specified file

Public Attributes

- string file
  Filename
- JObject json
  Json dictionary of settings

5.32.1 Detailed Description

Object representing settings json file

5.32.2 Constructor & Destructor Documentation

5.32.2.1 Settings()

```
RCCM.Settings.Settings (
            string filename )
```

Create settings object from file

Parameters

filename	Path to settings json file

5.32.3 Member Function Documentation

5.32.3.1 save() [1/2]

```
bool RCCM.Settings.save ( )
```

Save settings to original file they were read from

Returns

True if settings saved successfully, false otherwise

5.32.3.2 save() [2/2]

```
bool RCCM.Settings.save (
            string filename )
```

Save settings to a specified file

Parameters

filename	Path of file to save to

Returns

True if settings saved successfully, false otherwise

5.32.4 Member Data Documentation

5.32.4.1 file

```
string RCCM.Settings.file
```

Filename

5.32.4.2 json

```
JObject RCCM.Settings.json
```

Json dictionary of settings

The documentation for this class was generated from the following file:

- RCCM/Settings.cs

5.33 RCCM.TestResults Class Reference

Helper object for plotting test status such as crack lengths and pressure

Public Member Functions

- **TestResults** (RCCMSystem rccm, ObservableCollection< MeasurementSequence > cracks, Chart crackChart, Chart cycleChart, TextBox cycleIndicator, TextBox pressureIndicator, ListBox crackSelection)
- void PlotCracks ()

  Redraw charts

Protected Attributes

- ObservableCollection< MeasurementSequence > cracks

  List of all cracks being measured by RCCM cameras
- Chart crackChart

  Chart UI object for plotting crack length
- Chart cycleChart

  Chart UI object for plotting past pressure readings
- TextBox cycleIndicator

  Text indicator for cycle number
- TextBox pressureIndicator

  Text indicator for current pressure reading
- ListBox crackSelection

  Listbox with selectable crack names determining what to plot
- ICycleCounter counter

  RCCM cycle counter object
- Timer updateControlsTimer

  Timer for calling update function for UI
- int savedReadings

  Number of pressure readings to save and plot

5.33.1 Detailed Description

Helper object for plotting test status such as crack lengths and pressure

5.33.2 Member Function Documentation

5.33.2.1 PlotCracks()

```
void RCCM.TestResults.PlotCracks ( )
```

Redraw charts

5.33.3 Member Data Documentation

5.33.3.1 counter

```
ICycleCounter RCCM.TestResults.counter  [protected]
```

RCCM cycle counter object

5.33.3.2 crackChart

```
Chart RCCM.TestResults.crackChart  [protected]
```

Chart UI object for plotting crack length

5.33.3.3 cracks

```
ObservableCollection<MeasurementSequence> RCCM.TestResults.cracks  [protected]
```

List of all cracks being measured by RCCM cameras

5.33.3.4 crackSelection

```
ListBox RCCM.TestResults.crackSelection  [protected]
```

Listbox with selectable crack names determining what to plot

5.33.3.5 cycleChart

```
Chart RCCM.TestResults.cycleChart  [protected]
```

Chart UI object for plotting past pressure readings

5.33.3.6 cycleIndicator

```
TextBox RCCM.TestResults.cycleIndicator  [protected]
```

Text indicator for cycle number

5.33.3.7 pressureIndicator

```
TextBox RCCM.TestResults.pressureIndicator  [protected]
```

Text indicator for current pressure reading

5.33.3.8 savedReadings

```
int RCCM.TestResults.savedReadings  [protected]
```

Number of pressure readings to save and plot

5.33.3.9 updateControlsTimer

```
Timer RCCM.TestResults.updateControlsTimer  [protected]
```

Timer for calling update function for UI

The documentation for this class was generated from the following file:

- RCCM/TestResults.cs

5.34 RCCM.TrioController Class Reference

Class representing Trio stepper motor controller

Public Member Functions

- TrioController (AxTrioPC axTrioPC)

  Connect to and initialize Trio controller
- bool isMoving (short nAxis)

  Check if an axis is currently moving
- double [ ] GetAllAxisProperties (short nAxis)

  Get all axis property values
- double GetAxisProperty (string property, short nAxis)

  Get a specified property of a motor
- bool SetAxisProperty (string property, double value, short nAxis)

  Set a specified property of a motor
- double GetProperty (string property)

  Get property value (could be axis or controller property)
- bool MoveAbs (short nAxis, double pos)

  Move a specified actuator to a new coordinate
- bool MoveRel (short nAxis, double pos)

  Move a specified distance from current actuator position
- bool Jog (bool fwd, short nAxis)

  Begin moving an actuator continuously
- bool JogStop (short nAxis)

  Stop continuous actuator motion
- void Zero (short nAxis)

  Set current actuator position as 0 and clear errors
- bool Stop ()

  Stop all moving actuators
- void WaitForEndOfMove (short nAxis)

  Blocking function that completes once current actuator motion completes

Static Public Attributes

- static short PORT_TYPE = 2

  Enum constant identifying controller port as an ethernet connection
- static short PORT_ID = 3240

  Port id used to connect to controller
- static string IP = "192.168.0.250"

  Static IP address of controller
- static short NUMBER_AXES = 8

  Number of axes on controller
- static short ATYPE = 43

  Enum value identifying a step-direction stepper driver type
- static string [ ] AX_PROPERTIES = { "ATYPE", "P_GAIN", "I_GAIN", "D_GAIN", "OV_GAIN", "VFF_GAIN", "UNITS", "SPEED", "ACCEL", "DECEL", "CREEP", "JOGSPEED", "FE_LIMIT", "DAC", "SERVO", "REP_DIST", "FWD_IN", "REV_IN", "DATUM_IN", "FS_LIMIT", "RS_LIMIT", "MTYPE", "NTYPE", "MPOS", "DPOS", "FE", "AXISSTATUS" }

  Property names accessible for each motor

Properties

- bool Open `[get]`

  Indicates whether or not controller is connected and port is opened

5.34.1 Detailed Description

Class representing Trio stepper motor controller

5.34.2 Constructor & Destructor Documentation

5.34.2.1 TrioController()

```
RCCM.TrioController.TrioController (
            AxTrioPC axTrioPC )
```

Connect to and initialize Trio controller

Parameters

axTrioPC	Trio ActviveX control

5.34.3 Member Function Documentation

5.34.3.1 GetAllAxisProperties()

```
double [] RCCM.TrioController.GetAllAxisProperties (
            short nAxis )
```

Get all axis property values

Parameters

nAxis	Number (0-7) of port where axis is connected to trio controller

Returns

Array of property values with indices corresponding to TrioController.AX_PROPERTIES

5.34.3.2 GetAxisProperty()

```
double RCCM.TrioController.GetAxisProperty (
            string property,
            short nAxis )
```

Get a specified property of a motor

Parameters

property	Name of property to check
nAxis	Number (0-7) of port where axis is connected to trio controller

Returns

Specified property value

5.34.3.3 GetProperty()

```
double RCCM.TrioController.GetProperty (
            string property )
```

Get property value (could be axis or controller property)

Parameters

property	Property name

Returns

Current property value

5.34.3.4 isMoving()

```
bool RCCM.TrioController.isMoving (
            short nAxis )
```

Check if an axis is currently moving

Parameters

nAxis	Number (0-7) of port where axis is connected to trio controller

Returns

True if axis is performing a motion

5.34.3.5 Jog()

```
bool RCCM.TrioController.Jog (
            bool fwd,
            short nAxis )
```

Begin moving an actuator continuously

Parameters

fwd	Flag to indicate if actuator should move forward or backward
nAxis	Number (0-7) of port where axis is connected to trio controller

Returns

True if command was sent successfully

5.34.3.6 JogStop()

```
bool RCCM.TrioController.JogStop (
            short nAxis )
```

Stop continuous actuator motion

Parameters

nAxis	Number (0-7) of port where axis is connected to trio controller

Returns

True if command was sent successfully

5.34.3.7 MoveAbs()

```
bool RCCM.TrioController.MoveAbs (
            short nAxis,
            double pos )
```

Move a specified actuator to a new coordinate

Parameters

nAxis	Number (0-7) of port where axis is connected to trio controller
pos	New position of axis

Returns

True if command was sent successfully

5.34.3.8 MoveRel()

```
bool RCCM.TrioController.MoveRel (
            short nAxis,
            double pos )
```

Move a specified distance from current actuator position

Parameters

nAxis	Number (0-7) of port where axis is connected to trio controller
pos	Distance to move

Returns

True if command was sent successfully

5.34.3.9 SetAxisProperty()

```
bool RCCM.TrioController.SetAxisProperty (
            string property,
            double value,
            short nAxis )
```

Set a specified property of a motor

Parameters

property	Name of property to check
value	Newe value of property
nAxis	Number (0-7) of port where axis is connected to trio controller

Returns

True if value was set successfully

5.34.3.10 Stop()

```
bool RCCM.TrioController.Stop ( )
```

Stop all moving actuators

Returns

True if command was sent successfully

5.34.3.11 WaitForEndOfMove()

```
void RCCM.TrioController.WaitForEndOfMove (
            short nAxis )
```

Blocking function that completes once current actuator motion completes

Parameters

nAxis	Number (0-7) of port where axis is connected to trio controller

5.34.3.12 Zero()

```
void RCCM.TrioController.Zero (
            short nAxis )
```

Set current actuator position as 0 and clear errors

Parameters

nAxis	Number (0-7) of port where axis is connected to trio controller

5.34.4 Member Data Documentation

5.34.4.1 ATYPE

```
short RCCM.TrioController.ATYPE = 43  [static]
```

Enum value identifying a step-direction stepper driver type

5.34.4.2 AX_PROPERTIES

```
string [] RCCM.TrioController.AX_PROPERTIES = { "ATYPE", "P_GAIN", "I_GAIN", "D_GAIN", "OV_↩
GAIN", "VFF_GAIN", "UNITS", "SPEED", "ACCEL", "DECEL", "CREEP", "JOGSPEED", "FE_LIMIT", "DAC",
"SERVO", "REP_DIST", "FWD_IN", "REV_IN", "DATUM_IN", "FS_LIMIT", "RS_LIMIT", "MTYPE", "NTYPE",
"MPOS", "DPOS", "FE", "AXISSTATUS" }  [static]
```

Property names accessible for each motor

5.34.4.3 IP

```
string RCCM.TrioController.IP = "192.168.0.250"  [static]
```

Static IP address of controller

5.34.4.4 NUMBER_AXES

```
short RCCM.TrioController.NUMBER_AXES = 8  [static]
```

Number of axes on controller

5.34.4.5 PORT_ID

```
short RCCM.TrioController.PORT_ID = 3240  [static]
```

Port id used to connect to controller

5.34.4.6 PORT_TYPE

```
short RCCM.TrioController.PORT_TYPE = 2  [static]
```

Enum constant identifying controller port as an ethernet connection

5.34.5 Property Documentation

5.34.5.1 Open

```
bool RCCM.TrioController.Open  [get]
```

Indicates whether or not controller is connected and port is opened

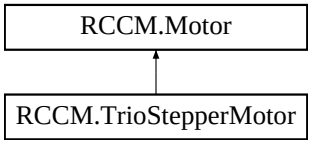
The documentation for this class was generated from the following file:

- RCCM/TrioController.cs

5.35 RCCM.TrioStepperMotor Class Reference

Object representing a physical motor controlled through Trio controller

Inheritance diagram for RCCM.TrioStepperMotor:



Public Member Functions

- TrioStepperMotor (TrioController controller, short axisNum)

  Create a trio motor object. Settings are not initialized in this function
- override void Jog (bool fwd)

  Begin moving this actuator continuously
- override void JogStop ()

  Stop moving axis continuously
- override bool Initialize ()

  Initialize status variables of superclass Motor
- override double SetPos (double cmd)

  Set the command position of the motor. After calling this method, getPos() will return cmd
- override double MoveRel (double dist)

  Set the command position of the motor. After calling this method, getPos() will return cmd
- override double GetPos ()

  Get motor position. This is accessed using the MPOS axis property
- override void Zero ()

  Set current actuator position as zero and clear errors
- override Dictionary< string, double > GetAllProperties ()

  Get all axis property values in a dictionary
- override double GetProperty (string property)

  Get a specified axis property
- override bool SetProperty (string property, double value)

  Set axis property
- override void WaitForEndOfMove ()

  Blocking function that completes when current actuator motion ends

Static Public Attributes

- static Dictionary< string, string > TRIO_PROPERTY_MAP

  Dictionary mapping settings file properties to corresponding Trio axis properties

Additional Inherited Members

5.35.1 Detailed Description

Object representing a physical motor controlled through Trio controller

5.35.2 Constructor & Destructor Documentation

5.35.2.1 TrioStepperMotor()

```
RCCM.TrioStepperMotor.TrioStepperMotor (
            TrioController controller,
            short axisNum )
```

Create a trio motor object. Settings are not initialized in this function

Parameters

controller	
axisNum	

5.35.3 Member Function Documentation

5.35.3.1 GetAllProperties()

```
override Dictionary<string, double> RCCM.TrioStepperMotor.GetAllProperties ( )  [virtual]
```

Get all axis property values in a dictionary

Returns

A dictionary of property name, value pairs

Implements RCCM.Motor.

5.35.3.2 GetPos()

```
override double RCCM.TrioStepperMotor.GetPos ( )  [virtual]
```

Get motor position. This is accessed using the MPOS axis property

Returns

Current position

Implements RCCM.Motor.

5.35.3.3 GetProperty()

```
override double RCCM.TrioStepperMotor.GetProperty (
            string property )  [virtual]
```

Get a specified axis property

Parameters

property	Property name

Returns

Current property value

Reimplemented from RCCM.Motor.

5.35.3.4 Initialize()

```
override bool RCCM.TrioStepperMotor.Initialize ( )  [virtual]
```

Initialize status variables of superclass Motor

Returns

Initialization status of motor

Implements RCCM.Motor.

5.35.3.5 Jog()

```
override void RCCM.TrioStepperMotor.Jog (
            bool fwd )  [virtual]
```

Begin moving this actuator continuously

Parameters

fwd	Flag indicating direction of motion. True corresponds to forward

Implements RCCM.Motor.

5.35.3.6 JogStop()

```
override void RCCM.TrioStepperMotor.JogStop ( )  [virtual]
```

Stop moving axis continuously

Implements RCCM.Motor.

5.35.3.7 MoveRel()

```
override double RCCM.TrioStepperMotor.MoveRel (
            double dist )  [virtual]
```

Set the command position of the motor. After calling this method, getPos() will return cmd

Parameters

dist	New command position

Returns

The previous commanded position

Implements RCCM.Motor.

5.35.3.8 SetPos()

```
override double RCCM.TrioStepperMotor.SetPos (
            double cmd )  [virtual]
```

Set the command position of the motor. After calling this method, getPos() will return cmd

Parameters

cmd	New command position

Returns

The new commanded position

Implements RCCM.Motor.

5.35.3.9 SetProperty()

```
override bool RCCM.TrioStepperMotor.SetProperty (
            string property,
            double value )  [virtual]
```

Set axis property

Parameters

property	Property name
value	New property value

Returns

True if property was set successfully

Reimplemented from RCCM.Motor.

5.35.3.10 WaitForEndOfMove()

```
override void RCCM.TrioStepperMotor.WaitForEndOfMove ( )  [virtual]
```

Blocking function that completes when current actuator motion ends

Implements RCCM.Motor.

5.35.3.11 Zero()

```
override void RCCM.TrioStepperMotor.Zero ( )  [virtual]
```

Set current actuator position as zero and clear errors

Reimplemented from RCCM.Motor.

5.35.4 Member Data Documentation

5.35.4.1 TRIO_PROPERTY_MAP

```
Dictionary<string, string> RCCM.TrioStepperMotor.TRIO_PROPERTY_MAP  [static]
```

Initial value:

```
= new Dictionary<string, string>{
            { "enabled", "AXIS_ENABLE" },
            { "microstep per mm", "UNITS" },
            { "velocity", "SPEED" },
            { "jog speed", "JOGSPEED" },
            { "acceleration", "ACCEL" },
            { "deceleration", "DECEL" },
            { "low position limit", "AXIS_RS_LIMIT" },
            { "high position limit", "AXIS_FS_LIMIT" },
            { "home", "" }
        }
```

Dictionary mapping settings file properties to corresponding Trio axis properties

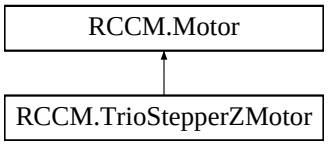
The documentation for this class was generated from the following file:

- RCCM/TrioStepperMotor.cs

5.36 RCCM.TrioStepperZMotor Class Reference

Actuator controlled through trio controller that adjusts its position based on distance sensor input

Inheritance diagram for RCCM.TrioStepperZMotor:



Public Member Functions

- TrioStepperZMotor (TrioController controller, short axisNum, RCCMSystem rccm, RCCMStage stage)

  Create a controlled z actuator
- override double GetPos ()

  Get actuator height above panel
- override double SetPos (double cmdHeight)

  Set current position of actuator (independent of feedback property value)
- override double MoveRel (double dist)

  Move command position a specified distance from current
- override bool Initialize ()

  Does nothing
- override Dictionary< string, double > GetAllProperties ()

  Get all property values in a dictionary
- override double GetProperty (string property)

  Get a specified property value
- override bool SetProperty (string property, double value)

  Set a property value
- override void WaitForEndOfMove ()

  Blocking function that completes when actuator motion ends
- override void Jog (bool fwd)

  Pauses active adjustment of actuator and begins moving axis continuously
- override void JogStop ()

  Stop jogging actuator and resume adjustment thread
- override void Zero ()

  Set current position of actuator as 0 and clear any errors
- override double GetActuatorPos ()

  Get position of actuator from end of travel
- void Pause ()

  Pause active height adjustment
- void Resume ()

  Unpause active height adjustment
- override void Terminate ()

  Stop background adjustment thread

Static Public Attributes

- static Dictionary< string, string > TRIO_PROPERTY_MAP

  Dictionary mapping settings property names to their corresponding Trio axis property
- static long UPDATE_PERIOD = (long)Program.Settings.json["distance sensor"]["z position update period"]

  Time in milliseconds between commands sent to controller
- static double ERROR = (double)Program.Settings.json["distance sensor"]["max height error"]

  Maximum allowable error between command height and measured height before axis is adjusted
- static double PGAIN = 0.5

  Gain multiplying position error to determine much correction actuator should do

Protected Attributes

- TrioController controller

  RCCM Trio controller object
- short axisNum

  Number of port where this axis is connected to Trio controller
- Func< double > height

  Function reference for getting height of actuator above panel
- Func< double > minPosition

  Function reference for computing lowest position that axis should move to from current position on panel
- double commandHeight

  Current desired height above panel
- BackgroundWorker bw

  Background worker for sending movement commands
- AutoResetEvent adjustThreadExited

  Event for when background thread exits
- bool adjust

  Flag indicating if background thread should continue running
- bool adjustThreadPaused

  Flag indicating if actuator should be actively trying to match height to command height

Additional Inherited Members

5.36.1 Detailed Description

Actuator controlled through trio controller that adjusts its position based on distance sensor input

5.36.2 Constructor & Destructor Documentation

5.36.2.1 TrioStepperZMotor()

```
RCCM.TrioStepperZMotor.TrioStepperZMotor (
            TrioController controller,
            short axisNum,
            RCCMSystem rccm,
            RCCMStage stage )
```

Create a controlled z actuator

Parameters

controller	RCCM Trio controller object
axisNum	Number of port where axis is connected to Trio controller
rccm	The RCCM object
stage	Enum value of the set of fine actuators containing this actuator

5.36.3 Member Function Documentation

5.36.3.1 GetActuatorPos()

```
override double RCCM.TrioStepperZMotor.GetActuatorPos ( )  [virtual]
```

Get position of actuator from end of travel

Returns

Current actuator position

Reimplemented from RCCM.Motor.

5.36.3.2 GetAllProperties()

```
override Dictionary<string, double> RCCM.TrioStepperZMotor.GetAllProperties ( )  [virtual]
```

Get all property values in a dictionary

Returns

Dictionary of property name, value pairs

Implements RCCM.Motor.

5.36.3.3 GetPos()

```
override double RCCM.TrioStepperZMotor.GetPos ( )  [virtual]
```

Get actuator height above panel

Returns

Current height of actuator above panel

Implements RCCM.Motor.

5.36.3.4 GetProperty()

```
override double RCCM.TrioStepperZMotor.GetProperty (
            string property )  [virtual]
```

Get a specified property value

Parameters

property	Name of property

Returns

Current property value

Reimplemented from RCCM.Motor.

5.36.3.5 Initialize()

```
override bool RCCM.TrioStepperZMotor.Initialize ( )  [virtual]
```

Does nothing

Returns

True if actuator is enabled

Implements RCCM.Motor.

5.36.3.6 Jog()

```
override void RCCM.TrioStepperZMotor.Jog (
            bool fwd )  [virtual]
```

Pauses active adjustment of actuator and begins moving axis continuously

Parameters

fwd	Flag indicating direction of move

Implements RCCM.Motor.

5.36.3.7 JogStop()

```
override void RCCM.TrioStepperZMotor.JogStop ( )  [virtual]
```

Stop jogging actuator and resume adjustment thread

Implements RCCM.Motor.

5.36.3.8 MoveRel()

```
override double RCCM.TrioStepperZMotor.MoveRel (
            double dist )  [virtual]
```

Move command position a specified distance from current

Parameters

dist	Distance to move

Returns

Last command position

Implements RCCM.Motor.

5.36.3.9 Pause()

```
void RCCM.TrioStepperZMotor.Pause ( )
```

Pause active height adjustment

5.36.3.10 Resume()

```
void RCCM.TrioStepperZMotor.Resume ( )
```

Unpause active height adjustment

5.36.3.11 SetPos()

```
override double RCCM.TrioStepperZMotor.SetPos (
            double cmdHeight )  [virtual]
```

Set current position of actuator (independent of feedback property value)

Parameters

cmdHeight	New command position of actuator

Returns

Coerced command position

Implements RCCM.Motor.

5.36.3.12 SetProperty()

```
override bool RCCM.TrioStepperZMotor.SetProperty (
            string property,
            double value )  [virtual]
```

Set a property value

Parameters

property	Property name
value	New property value

Returns

True if property was set succesfully

Reimplemented from RCCM.Motor.

5.36.3.13 Terminate()

```
override void RCCM.TrioStepperZMotor.Terminate ( )  [virtual]
```

Stop background adjustment thread

Reimplemented from RCCM.Motor.

5.36.3.14 WaitForEndOfMove()

```
override void RCCM.TrioStepperZMotor.WaitForEndOfMove ( )  [virtual]
```

Blocking function that completes when actuator motion ends

Implements RCCM.Motor.

5.36.3.15 Zero()

```
override void RCCM.TrioStepperZMotor.Zero ( )  [virtual]
```

Set current position of actuator as 0 and clear any errors

Reimplemented from RCCM.Motor.

5.36.4 Member Data Documentation

5.36.4.1 adjust

```
bool RCCM.TrioStepperZMotor.adjust  [protected]
```

Flag indicating if background thread should continue running

5.36.4.2 adjustThreadExited

```
AutoResetEvent RCCM.TrioStepperZMotor.adjustThreadExited  [protected]
```

Event for when background thread exits

5.36.4.3 adjustThreadPaused

```
bool RCCM.TrioStepperZMotor.adjustThreadPaused  [protected]
```

Flag indicating if actuator should be actively trying to match height to command height

5.36.4.4 axisNum

```
short RCCM.TrioStepperZMotor.axisNum  [protected]
```

Number of port where this axis is connected to Trio controller

5.36.4.5 bw

```
BackgroundWorker RCCM.TrioStepperZMotor.bw  [protected]
```

Background worker for sending movement commands

5.36.4.6 commandHeight

```
double RCCM.TrioStepperZMotor.commandHeight  [protected]
```

Current desired height above panel

5.36.4.7 controller

```
TrioController RCCM.TrioStepperZMotor.controller  [protected]
```

RCCM Trio controller object

5.36.4.8 ERROR

```
double RCCM.TrioStepperZMotor.ERROR = (double)Program.Settings.json["distance sensor"]["max
height error"]  [static]
```

Maximum allowable error between command height and measured height before axis is adjusted

5.36.4.9 height

```
Func<double> RCCM.TrioStepperZMotor.height  [protected]
```

Function reference for getting height of actuator above panel

5.36.4.10 minPosition

```
Func<double> RCCM.TrioStepperZMotor.minPosition  [protected]
```

Function reference for computing lowest position that axis should move to from current position on panel

5.36.4.11 PGAIN

```
double RCCM.TrioStepperZMotor.PGAIN = 0.5  [static]
```

Gain multiplying position error to determine much correction actuator should do

5.36.4.12 TRIO_PROPERTY_MAP

```
Dictionary<string, string> RCCM.TrioStepperZMotor.TRIO_PROPERTY_MAP  [static]
```

Initial value:

```
= new Dictionary<string, string>{
            { "enabled", "AXIS_ENABLE" },
            { "microstep per mm", "UNITS" },
            { "velocity", "SPEED" },
            { "jog speed", "JOGSPEED" },
            { "acceleration", "ACCEL" },
            { "deceleration", "DECEL" },
            { "low position limit", "AXIS_RS_LIMIT" },
            { "high position limit", "AXIS_FS_LIMIT" },
            { "home", "" },
            { "feedback", "" }
        }
```

Dictionary mapping settings property names to their corresponding Trio axis property

5.36.4.13 UPDATE_PERIOD

```
long RCCM.TrioStepperZMotor.UPDATE_PERIOD = (long)Program.Settings.json["distance sensor"]["z position update period"]  [static]
```

Time in milliseconds between commands sent to controller

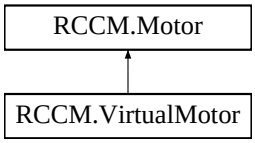
The documentation for this class was generated from the following file:

- RCCM/TrioStepperZMotor.cs

5.37 RCCM.VirtualMotor Class Reference

Virtual representation of motor for use when motor is not connected

Inheritance diagram for RCCM.VirtualMotor:



Public Member Functions

- override double GetPos ()

  Get motor position. For a virtual motor, this is simply the last commanded position
- override double SetPos (double cmd)

  Set the command position of the motor. After calling this method, getPos() will return cmd
- override double MoveRel (double dist)

  Move the motor a specified distance
- override bool Initialize ()

  Initialize status variables of superclass Motor
- override Dictionary< string, double > GetAllProperties ()

  Get all property value pairs for this motor in a dictionary
- override void WaitForEndOfMove ()

  Since move is applied instantaneously, no waiting necessary
- override void Jog (bool fwd)

  Pretend to jog this motor lol
- override void JogStop ()

  Pretend to stop jogging this motor lol

Additional Inherited Members

5.37.1 Detailed Description

Virtual representation of motor for use when motor is not connected

5.37.2 Member Function Documentation

5.37.2.1 GetAllProperties()

```
override Dictionary<string, double> RCCM.VirtualMotor.GetAllProperties ( )  [virtual]
```

Get all property value pairs for this motor in a dictionary

Returns

Dictionary of property value pairs

Implements RCCM.Motor.

5.37.2.2 GetPos()

```
override double RCCM.VirtualMotor.GetPos ( )  [virtual]
```

Get motor position. For a virtual motor, this is simply the last commanded position

Returns

Current position

Implements RCCM.Motor.

5.37.2.3 Initialize()

```
override bool RCCM.VirtualMotor.Initialize ( )  [virtual]
```

Initialize status variables of superclass Motor

Returns

Initialization status of motor

Implements RCCM.Motor.

5.37.2.4 Jog()

```
override void RCCM.VirtualMotor.Jog (
            bool fwd )  [virtual]
```

Pretend to jog this motor lol

Parameters

fwd	Direction of jog - forward if true

Implements RCCM.Motor.

5.37.2.5 JogStop()

```
override void RCCM.VirtualMotor.JogStop ( )  [virtual]
```

Pretend to stop jogging this motor lol

Implements RCCM.Motor.

5.37.2.6 MoveRel()

```
override double RCCM.VirtualMotor.MoveRel (
            double dist )  [virtual]
```

Move the motor a specified distance

Parameters

dist	Distance to move

Returns

The previous commanded position

Implements RCCM.Motor.

5.37.2.7 SetPos()

```
override double RCCM.VirtualMotor.SetPos (
            double cmd )  [virtual]
```

Set the command position of the motor. After calling this method, getPos() will return cmd

Parameters

cmd	New command position

Returns

The new commanded position

Implements RCCM.Motor.

5.37.2.8 WaitForEndOfMove()

```
override void RCCM.VirtualMotor.WaitForEndOfMove ( )  [virtual]
```

Since move is applied instantaneously, no waiting necessary

Implements RCCM.Motor.

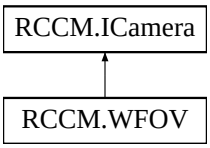
The documentation for this class was generated from the following file:

- RCCM/VirtualMotor.cs

5.38 RCCM.WFOV Class Reference

Class representing DMK Z12G445 camera for the RCCM WFOV

Inheritance diagram for RCCM.WFOV:



Public Member Functions

- WFOV (string name)
  Create WFOV camera from configuration file
- bool Initialize (ICImagingControl ic)
  Connect to camera. Will fail if configuration file referred to invalid or disconnected camera
- void Start ()
  Begin displaying live image
- void Stop ()
  Stop displaying live image. Will also cease video recording, if active
- void Snap (string filename)
  Capture live image to file
- void Record (string filename)
  Start recording video to specified path
- void StopRecord ()
  Stop recording video. Will resume live display after stopping recording
- void SetScale (RCCMSystem rccm, double scale)
  Set image scale and save current height, zoom, and focus
- bool CheckFOV (RCCMSystem rccm)
  Check if measurement conditions match calibration conditions
- void EditProperties ()
  Show device property dialog <warning>Will cause device to crash if "Cancel" button is pressed from property dialog</warning>
- int AutoFocus ()
  Activate built-in camera autofocus. Requires 2 second sleep to allow autofocus to complete

Static Public Attributes

- static int IMG_HEIGHT = 960

  Height in pixels of image
- static int IMG_WIDTH = 1280

  Width in pixels of image

Protected Attributes

- ICImagingControl ic

  Imaging user control for displaying the live image
- VCDSimpleProperty VCDProp

  Camera properties accessor

Properties

- bool Available `[get]`

  Flag to indicate connection status of WFOV camera
- double Scale `[get, protected set]`

  Camera microns / pixel calibration
- double Height `[get]`

  Height in mm of image
- double Width `[get]`

  Height in mm of image
- int Zoom `[get, set]`

  Zoom level of the camera, an integer between 0 and 100
- int Focus `[get, set]`

  Focal distance of the camera, roughly equating to distance in mm
- int FocusMin `[get]`

  Minimum focal distance of the camera
- int FocusMax `[get]`

  Maximum focal distance of the camera
- int ZoomMin `[get]`

  Minimum zoom level of the camera
- int ZoomMax `[get]`

  Maximum zoom level of the camera
- bool Recording `[get]`

  Flag to indicate if a video is being recorded
- string configFile `[get, set]`

  File path to configuration file from which camera is initialized
- double CalibrationHeight `[get, protected set]`

  File path to configuration file from which camera is initialized
- double CalibrationZoom `[get, protected set]`

  File path to configuration file from which camera is initialized
- double CalibrationFocus `[get, protected set]`

  File path to configuration file from which camera is initialized

5.38.1 Detailed Description

Class representing DMK Z12G445 camera for the RCCM WFOV

5.38.2 Constructor & Destructor Documentation

5.38.2.1 WFOV()

```
RCCM.WFOV.WFOV (
            string name )
```

Create WFOV camera from configuration file

Parameters

name	Name of camera in settings

5.38.3 Member Function Documentation

5.38.3.1 AutoFocus()

```
int RCCM.WFOV.AutoFocus ( )
```

Activate built-in camera autofocus. Requires 2 second sleep to allow autofocus to complete

Returns

New focus level

5.38.3.2 CheckFOV()

```
bool RCCM.WFOV.CheckFOV (
            RCCMSystem rccm )
```

Check if measurement conditions match calibration conditions

Parameters

rccm	

Returns

True if measurement conditions match calibration

Implements RCCM.ICamera.

5.38.3.3 EditProperties()

```
void RCCM.WFOV.EditProperties ( )
```

Show device property dialog <warning>Will cause device to crash if "Cancel" button is pressed from property dialog</warning>

5.38.3.4 Initialize()

```
bool RCCM.WFOV.Initialize (
            ICImagingControl ic )
```

Connect to camera. Will fail if configuration file referred to invalid or disconnected camera

Returns

True if initialization is successful

5.38.3.5 Record()

```
void RCCM.WFOV.Record (
            string filename )
```

Start recording video to specified path

Parameters

filename	

Implements RCCM.ICamera.

5.38.3.6 SetScale()

```
void RCCM.WFOV.SetScale (
            RCCMSystem rccm,
            double scale )
```

Set image scale and save current height, zoom, and focus

Parameters

rccm	
scale	New calibration

Implements RCCM.ICamera.

5.38.3.7 Snap()

```
void RCCM.WFOV.Snap (
            string filename )
```

Capture live image to file

Parameters

filename	Filename to save image to. Should have .png extension

Implements RCCM.ICamera.

5.38.3.8 Start()

```
void RCCM.WFOV.Start ( )
```

Begin displaying live image

Implements RCCM.ICamera.

5.38.3.9 Stop()

```
void RCCM.WFOV.Stop ( )
```

Stop displaying live image. Will also cease video recording, if active

Implements RCCM.ICamera.

5.38.3.10 StopRecord()

```
void RCCM.WFOV.StopRecord ( )
```

Stop recording video. Will resume live display after stopping recording

Implements RCCM.ICamera.

5.38.4 Member Data Documentation

5.38.4.1 ic

```
ICImagingControl RCCM.WFOV.ic  [protected]
```

Imaging user control for displaying the live image

5.38.4.2 IMG_HEIGHT

```
int RCCM.WFOV.IMG_HEIGHT = 960  [static]
```

Height in pixels of image

5.38.4.3 IMG_WIDTH

```
int RCCM.WFOV.IMG_WIDTH = 1280  [static]
```

Width in pixels of image

5.38.4.4 VCDProp

```
VCDSimpleProperty RCCM.WFOV.VCDProp  [protected]
```

Camera properties accessor

5.38.5 Property Documentation

5.38.5.1 Available

```
bool RCCM.WFOV.Available  [get]
```

Flag to indicate connection status of WFOV camera

5.38.5.2 CalibrationFocus

```
double RCCM.WFOV.CalibrationFocus  [get], [protected set]
```

File path to configuration file from which camera is initialized

5.38.5.3 CalibrationHeight

```
double RCCM.WFOV.CalibrationHeight  [get], [protected set]
```

File path to configuration file from which camera is initialized

5.38.5.4 CalibrationZoom

```
double RCCM.WFOV.CalibrationZoom  [get], [protected set]
```

File path to configuration file from which camera is initialized

5.38.5.5 configFile

```
string RCCM.WFOV.configFile  [get], [set]
```

File path to configuration file from which camera is initialized

5.38.5.6 Focus

```
int RCCM.WFOV.Focus  [get], [set]
```

Focal distance of the camera, roughly equating to distance in mm

5.38.5.7 FocusMax

```
int RCCM.WFOV.FocusMax  [get]
```

Maximum focal distance of the camera

5.38.5.8 FocusMin

```
int RCCM.WFOV.FocusMin  [get]
```

Minimum focal distance of the camera

5.38.5.9 Height

```
double RCCM.WFOV.Height  [get]
```

Height in mm of image

5.38.5.10 Recording

```
bool RCCM.WFOV.Recording  [get]
```

Flag to indicate if a video is being recorded

5.38.5.11 Scale

```
double RCCM.WFOV.Scale  [get], [protected set]
```

Camera microns / pixel calibration

5.38.5.12 Width

```
double RCCM.WFOV.Width  [get]
```

Height in mm of image

5.38.5.13 Zoom

```
int RCCM.WFOV.Zoom  [get], [set]
```

Zoom level of the camera, an integer between 0 and 100

5.38.5.14 ZoomMax

```
int RCCM.WFOV.ZoomMax  [get]
```

Maximum zoom level of the camera

5.38.5.15 ZoomMin

```
int RCCM.WFOV.ZoomMin  [get]
```

Minimum zoom level of the camera

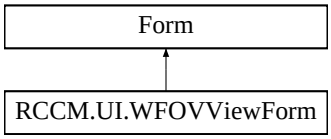
The documentation for this class was generated from the following file:

- RCCM/WFOV.cs

5.39 RCCM.UI.WFOVViewForm Class Reference

Form for displaying WFOV live image and measurement overlay

Inheritance diagram for RCCM.UI.WFOVViewForm:



Public Member Functions

- WFOVViewForm (RCCMSystem rccm, WFOV camera, ObservableCollection< MeasurementSequence > cracks)

  Create form and initialize given camera
- void createSegment ()

  Create segment that was being drawn by user with mouse input. This will add a point or segment to the active MeasurementSequence
- void moveDrawnLineEnd (int x, int y, int w, int h)

  Move the end point of the line segment that user is currently drawing
- void createDrawnLine (int x, int y, int w, int h)

  Begin drawing new segment in active MeasurementSequence from user mouse input
- void createPoint (int x, int y, int w, int h)

  Create new point in active measurement sequence at mouse location

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

Protected Attributes

- readonly RCCMSystem rccm

  The RCCM system object. Used for calculating WFOV position
- readonly ObservableCollection< MeasurementSequence > cracks

  List of cracks that are being measured
- readonly WFOV camera

  WFOV camera that is displayed by this window
- RCCMStage stage

  Parent stage of this camera
- PointF drawnLineStart

  Point where line user is drawing begins
- PointF drawnLineEnd

  Point where line user is drawing ends

Properties

- int ActiveIndex `[get, set]`

  Currently selected crack index. The selected crack will be edited by other controls
- int ActivePoint `[get, set]`

  Currently selected point in ListPoints. This point is indicated in the image display
- float displayScale `[get]`

  Ratio of picture box size to actual NFOV image dimensions
- bool Drawing `[get]`

  Indicates whether or not user is drawing a line with mouse

5.39.1 Detailed Description

Form for displaying WFOV live image and measurement overlay

5.39.2 Constructor & Destructor Documentation

5.39.2.1 WFOVViewForm()

```
RCCM.UI.WFOVViewForm.WFOVViewForm (
            RCCMSystem rccm,
            WFOV camera,
            ObservableCollection< MeasurementSequence > cracks )
```

Create form and initialize given camera

Parameters

rccm	Reference to the RCCM object
camera	Camera to be displayed
cracks	Reference to the list of all measurement sequences

5.39.3 Member Function Documentation

5.39.3.1 createDrawnLine()

```
void RCCM.UI.WFOVViewForm.createDrawnLine (
            int x,
            int y,
            int w,
            int h )
```

Begin drawing new segment in active MeasurementSequence from user mouse input

Parameters

x	Mouse x location in pixels
y	Mouse y location in pixels
w	Canvas width in pixels
h	Canvas height in pixels

5.39.3.2 createPoint()

```
void RCCM.UI.WFOVViewForm.createPoint (
            int x,
            int y,
            int w,
            int h )
```

Create new point in active measurement sequence at mouse location

Parameters

x	Mouse x location in pixels
y	Mouse y location in pixels
w	Canvas width in pixels
h	Canvas height in pixels

5.39.3.3 createSegment()

```
void RCCM.UI.WFOVViewForm.createSegment ( )
```

Create segment that was being drawn by user with mouse input. This will add a point or segment to the active MeasurementSequence

5.39.3.4 Dispose()

```
override void RCCM.UI.WFOVViewForm.Dispose (
            bool disposing )  [protected]
```

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

5.39.3.5 moveDrawnLineEnd()

```
void RCCM.UI.WFOVViewForm.moveDrawnLineEnd (
            int x,
            int y,
            int w,
            int h )
```

Move the end point of the line segment that user is currently drawing

Parameters

x	Mouse x location in pixels
y	Mouse y location in pixels
w	Canvas width in pixels
h	Canvas height in pixels

5.39.4 Member Data Documentation

5.39.4.1 camera

```
readonly WFOV RCCM.UI.WFOVViewForm.camera  [protected]
```

WFOV camera that is displayed by this window

5.39.4.2 cracks

```
readonly ObservableCollection<MeasurementSequence> RCCM.UI.WFOVViewForm.cracks  [protected]
```

List of cracks that are being measured

5.39.4.3 drawnLineEnd

```
PointF RCCM.UI.WFOVViewForm.drawnLineEnd  [protected]
```

Point where line user is drawing ends

5.39.4.4 drawnLineStart

```
PointF RCCM.UI.WFOVViewForm.drawnLineStart  [protected]
```

Point where line user is drawing begins

5.39.4.5 rccm

```
readonly RCCMSystem RCCM.UI.WFOVViewForm.rccm  [protected]
```

The RCCM system object. Used for calculating WFOV position

5.39.4.6 stage

```
RCCMStage RCCM.UI.WFOVViewForm.stage  [protected]
```

Parent stage of this camera

5.39.5 Property Documentation

5.39.5.1 ActiveIndex

```
int RCCM.UI.WFOVViewForm.ActiveIndex  [get], [set], [protected]
```

Currently selected crack index. The selected crack will be edited by other controls

5.39.5.2 ActivePoint

```
int RCCM.UI.WFOVViewForm.ActivePoint  [get], [set], [protected]
```

Currently selected point in ListPoints. This point is indicated in the image display

5.39.5.3 displayScale

```
float RCCM.UI.WFOVViewForm.displayScale  [get], [protected]
```

Ratio of picture box size to actual NFOV image dimensions

5.39.5.4 Drawing

```
bool RCCM.UI.WFOVViewForm.Drawing  [get], [protected]
```

Indicates whether or not user is drawing a line with mouse

The documentation for this class was generated from the following files:

- RCCM/UI/WFOVViewForm.cs
- RCCM/UI/WFOVViewForm.Designer.cs