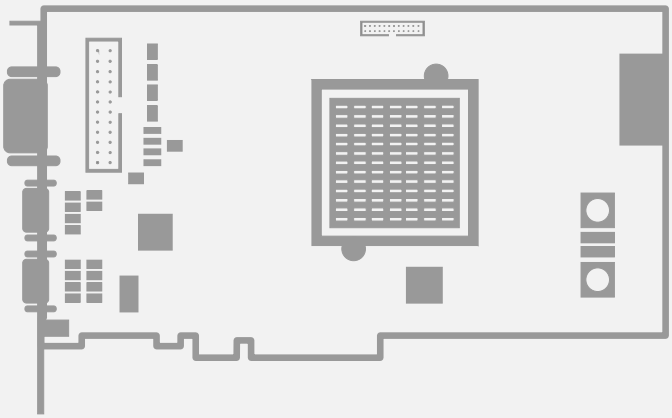
REFERENCE

Grablink

Grablink Parameters

1622 Grablink Full
1623 Grablink DualBase
1624 Grablink Base
1626 Grablink Full XR



Contents

1. About This Document

1.1. Scope and Summary

This document is a filtered edition of the MultiCam parameters reference for Grablink products supported by MultiCam driver version **6.19**:

Grablink main products

Product	S/N Prefix	Icon
1622 Grablink Full	FM1	Full
1623 Grablink DualBase	GDB	DualBase
1624 Grablink Base	GBA	Base
1626 Grablink Full XR	FXR	FullXR

Parameters are grouped by MultiCam object class. Classes are listed in the top-down hierarchical order. Within each class, parameters are listed in the natural order and grouped by categories.

The main sections of the document are:

- "Configuration Class" on page 12 : parameters of the MultiCam Configuration object
- "Board Class" on page 25: parameters of the MultiCam Board object
- "Channel Class" on page 84: parameters of the MultiCam Channel object.
- "Surface Class" on page 463: parameters of the MultiCam Surface object.
- "Annex" on page 483: selection of topics referenced in this document.

1.2. Document Changes

MultiCam 6.18

The following topics were added:

- "FifoOrderingYTapCount" on page 444

The following topics were revised:

- "FifoOrdering" on page 442
- "TapConfiguration" on page 109
- "CC3Usage" on page 191

2. Configuration Class

What Is the Configuration Object?

The Configuration object groups all MultiCam parameters dedicated to the control of system wide features.

The system should be basically understood as the set of Euresys boards installed inside a host computer. The configuration object also addresses any hardware or software element of the host computer requesting some degree of control for the MultiCam system operation.

The configuration object does not belong to a true class, as it is unique within the system. There is no need for the user to instantiate a Configuration class object using the McCreate or McCreateNm function. The Configuration object is natively made available to the application when the MultiCam driver is connected to it.

2.1. Configuration Category

Parameters specifying system wide features

MementoCritical

Sends Memento trace with a Critical level from user application

Parameter Info

Class	Category	Level	Type	Access
Configuration	Configuration	SELECT	String collection	Set Only

Num ID	String Identifier	C, C++ identifier
102 << 14	MementoCritical	MC_MementoCritical

Parameter Description

This string collection parameter of 16 elements enables the caller to send a Memento trace with "Critical" level from a user application.

Parameter Usage

The collection element index selects the Memento Kind from User0 to UserF.

For instance, to send a critical message with the User7 kind, the following call will be added in the C/C++ user application:

```
McSetParamStr(MC_CONFIGURATION, MC_MementoCritical + 7, "This is a critical message.");
```

MementoError

Sends Memento trace with an Error level from user application

Parameter Info

Class	Category	Level	Type	Access
Configuration	Configuration	SELECT	String collection	Set Only

Num ID	String Identifier	C, C++ identifier
103 << 14	MementoError	MC_MementoError

Parameter Description

This string collection parameter of 16 elements enables the caller to send a Memento trace with "Error" level from a user application.

Parameter Usage

The collection element index selects the Memento Kind from User0 to UserF.

For instance, to send an error message with the User7 kind, the following call will be added in the C/C++ user application:

```
McSetParamStr(MC_CONFIGURATION, MC_MementoError + 7, "This is an error message.");
```

MementoWarning

Sends Memento trace with a Warning level from user application

Parameter Info

Class	Category	Level	Type	Access
Configuration	Configuration	SELECT	String collection	Set Only

Num ID	String Identifier	C, C++ identifier
104 << 14	MementoWarning	MC_MementoWarning

Parameter Description

This string collection parameter of 16 elements enables the caller to send a Memento trace with "Warning" level from a user application.

Parameter Usage

The collection element index selects the Memento Kind from User0 to UserF.

For instance, to send a warning message with the User7 kind, the following call will be added in the C/C++ user application:

```
McSetParamStr(MC_CONFIGURATION, MC_MementoWarning + 7, "This is a warning message.");
```

MementoNotice

Sends Memento trace with a Notice level from user application

Parameter Info

Class	Category	Level	Type	Access
Configuration	Configuration	SELECT	String collection	Set Only

Num ID	String Identifier	C, C++ identifier
105 << 14	MementoNotice	MC_MementoNotice

Parameter Description

This string collection parameter of 16 elements enables the caller to send a Memento trace with "Notice" level from a user application.

Parameter Usage

The collection element index selects the Memento Kind from User0 to UserF.

For instance, to send a notice message with the User7 kind, the following call will be added in the C/C++ user application:

```
McSetParamStr(MC_CONFIGURATION, MC_MementoNotice + 7, "This is a notice message.");
```

MementoInfo

Sends Memento trace with an Info level from user application

Parameter Info

Class	Category	Level	Type	Access
Configuration	Configuration	SELECT	String collection	Set Only

Num ID	String Identifier	C, C++ identifier
106 << 14	MementoInfo	MC_MementoInfo

Parameter Description

This string collection parameter of 16 elements enables the caller to send a Memento trace with "Info" level from a user application.

Parameter Usage

The collection element index selects the Memento Kind from User0 to UserF.

For instance, to send an information message with the User7 kind, the following call will be added in the C/C++ user application:

```
McSetParamStr(MC_CONFIGURATION, MC_MementoInfo + 7, "This is an information message.");
```

MementoDebug

Sends Memento trace with a Debug level from user application

Parameter Info

Class	Category	Level	Type	Access
Configuration	Configuration	SELECT	String collection	Set Only

Num ID	String Identifier	C, C++ identifier
107 << 14	MementoDebug	MC_MementoDebug

Parameter Description

This string collection parameter of 16 elements enables the caller to send a Memento trace with "Debug" level from a user application.

Parameter Usage

The collection element index selects the Memento Kind from User0 to UserF.

For instance, to send a debug message with the User7 kind, the following call will be added in the C/C++ user application:

```
McSetParamStr(MC_CONFIGURATION, MC_MementoDebug + 7, "This is a debug message.");
```

MementoVerbose

Sends Memento trace with a Verbose level from user application

Parameter Info

Class	Category	Level	Type	Access
Configuration	Configuration	SELECT	String collection	Set Only

Num ID	String Identifier	C, C++ identifier
108 << 14	MementoVerbose	MC_MementoVerbose

Parameter Description

This string collection parameter of 16 elements enables the caller to send a Memento trace with "Verbose" level from a user application.

Parameter Usage

The collection element index selects the Memento Kind from User0 to UserF.

For instance, to send a verbose message with the User7 kind, the following call will be added in the C/C++ user application:

```
McSetParamStr(MC_CONFIGURATION, MC_MementoVerbose + 7, "This is a verbose message.");
```

BoardCount

Number of MultiCam boards in the system

Parameter Info

Class	Category	Level	Type	Access
Configuration	Configuration	ADJUST	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
62 << 14	BoardCount	MC_BoardCount

Parameter Description

This parameter provides an immediate way for the application to be informed on the number of peripheral boards recognized as MultiCam compliant boards.

See also "Code Example: How to Gather Board Information?" on page 506

ErrorHandling

Error handling behavior definition

Parameter Info

Class	Category	Level	Type	Access
Configuration	Configuration	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
49 << 14	ErrorHandling	MC_ErrorHandling

Parameter Description

This parameter defines the error handling behavior.

Parameter Usage

Directive: When operating with Windows, select any of the four available behaviors when an error occurs during the execution of a MultiCam API function.

Directive: When operating with Linux,leave the default value.

Parameter Values

NONE

MC_ErrorHandling_NONE
Description On error, the MultiCam driver returns an error code.
Default value.

MSGBOX

MC_ErrorHandling_MSGBOX
Description On error, the MultiCam driver displays an error dialog box then returns an error code.

EXCEPTION

MC_ErrorHandling_EXCEPTION
Description On error, the MultiCam driver issues a Windows structured exception.

MSGEXCEPTION

MC_ErrorHandling_MSGEXCEPTION

Description
On error, the MultiCam driver displays an error dialog box then issues a Windows structured exception.

ErrorLog

Path and filename of the error log file

Parameter Info

Class	Category	Level	Type	Access
Configuration	Configuration	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
81 << 14	ErrorLog	MC_ErrorLog

Parameter Description

This parameter specifies the path and the filename of the error log file that is created when the application returns a MC_INVALID_PARAMETER_SETTING (-22) error code.

The incorrect parameters are reported in the log file, including the wrong value and the possible correct values.

When specified, the log file is created and filled during the consistency check.

When unspecified, the consistency check does not produce a log file.

3. Board Class

What Is the Board Object?

The Board object groups all MultiCam parameters dedicated to the control of features specific to a board.

The Board object MultiCam parameters also address the access of I/O lines from an application program, implementing the general-purpose I/O functionality.

The Board object does not belong to a true class, as it is unique for each Euresys board installed inside a host computer. There is no need for the user to instantiate a Board class object using the McCreate or McCreateNm function. The Board objects are natively made available to the application for each installed Euresys board when the MultiCam driver is opened.

3.1. Board Information Category

Parameters providing access to identification, structure or security features of the board

BoardTopology

Arrangement of the cameras connected to the board and features set

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	SELECT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
59 << 14	BoardTopology	MC_BoardTopology

Parameter Description

This parameter defines the arrangement of cameras that can be potentially connected to the frame grabber.

When multiple feature sets are available for a board, it allows to select the appropriate feature set for the application.

Parameter Usage

Directive: The application must set this parameter before the first assignation of a MultiCam Channel to this board; it must not be modified while at least one channel is assigned to the board.

Directive: The parameter value can be modified only if no Camera Link serial port is in use. Otherwise, the MultiCam driver will return an MC_IO_ERROR status.

Parameter Values

MONO

Base Full FullXR

MC_BoardTopology_MONO

Base

Description
One Camera Link Base or one Camera Link Lite camera attached to the CAMERA connector.

Full FullXR

Description
One Camera Link Base camera attached to the BASE connector or ...
one Camera Link Medium or one Camera Link Full camera attached to the BASE and MEDIUM/FULL connectors.

> NOTE
> Do not use for Camera Link clock frequencies below 30 MHz!

Default value.

MONO_OPT1

Base Full FullXR

MC_BoardTopology_MONO_OPT1

Base

Description
One Camera Link Base or one Camera Link Lite camera attached to the CAMERA connector.
Includes the pipeline controller optional feature.

Full FullXR

Description
One Camera Link Base camera attached to the BASE connector or ...
one Camera Link Medium or one Camera Link Full camera attached to the BASE and MEDIUM/FULL connectors.
Includes the pipeline controller optional feature.

> NOTE
> Do not use for Camera Link clock frequencies below 30 MHz!

MONO_DECA

Full FullXR

MC_BoardTopology_MONO_DECA

Description
One Camera Link 72 Bit camera or one Camera Link 80 Bit camera attached to the BASE and MEDIUM/FULL connectors.

MONO_DECA_OPT1

Full FullXR

MC_BoardTopology_MONO_DECA_OPT1

Description
One Camera Link 72 Bit camera or one Camera Link 80 Bit camera attached to the BASE and MEDIUM/FULL connectors.
Includes the pipeline controller optional feature.

MONO_SLOW

Base Full FullXR

MC_BoardTopology_MONO_SLOW

Base

Description
One Camera Link Base or one Camera Link Lite camera attached to the CAMERA connector.
The cable deskewing function of the Camera Link interface is turned off.

NOTE
This setting is mandatory for Camera Link clock frequencies below 30 MHz.

Full FullXR

Description
One Camera Link Base camera attached to the BASE connector or ...
one Camera Link Medium or one Camera Link Full camera attached to the BASE and MEDIUM/FULL connectors.
The cable deskewing function of the Camera Link interface is turned off.

NOTE
This setting is mandatory for Camera Link clock frequencies below 30 MHz.

DUO

DualBase

MC_BoardTopology_DUO

Description
One Camera Link Base or one Camera Link Lite camera attached to the A connector and ...
one Camera Link Base or one Camera Link Lite camera attached to the B connector.

NOTE
Do not use for Camera Link clock frequencies below 30 MHz!

Default value.

DUO_OPT1

DualBase

MC_BoardTopology_DUO_OPT1

Description
One Camera Link Base or one Camera Link Lite camera attached to the A connector and ...
one Camera Link Base or one Camera Link Lite camera attached to the B connector.
Includes the pipeline controller optional feature.

NOTE
Do not use for Camera Link clock frequencies below 30 MHz!

DUO_SLOW

DualBase

MC_BoardTopology_DUO_SLOW

Description
One Camera Link Base or one Camera Link Lite camera attached to the A connector and ...
one Camera Link Base or one Camera Link Lite camera attached to the B connector.
The cable deskewing function of the Camera Link interface is turned off.

NOTE
This setting is mandatory for Camera Link clock frequencies below 30 MHz.

SerialLinkA

Base DualBase

Serial COM receiver source of Camera connector A or M

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	SELECT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10565 << 14	SerialLinkA	MC_SerialLinkA

Parameter Description

Selects the receiver source of the serial COM of the first Camera connector.

Parameter Usage

Directive: Set POCL_LITE when attaching a PoCL-Lite camera.

Parameter Values

STANDARD

Base DualBase

MC_SerialLinkA_STANDARD
Description The camera-to-frame-grabber serial communication link uses a dedicated line of the standard Camera Link cable
Default value.

POCL_LITE

Base DualBase

MC_SerialLinkA_POCL_LITE
Description The camera-to-frame-grabber serial link is embedded in the Channel Link of PoCL-Lite Camera Link cables

SerialLinkB

DualBase

Serial COM receiver source of Camera connector B

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	SELECT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10566 << 14	SerialLinkB	MC_SerialLinkB

Parameter Description

Selects the receiver source of the serial COM of the second Camera connector.

Parameter Usage

Directive: Set POCL_LITE when attaching a PoCL-Lite camera.

Parameter Values

STANDARD

DualBase

MC_SerialLinkB_STANDARD
Description The camera-to-frame-grabber serial communication link uses a dedicated line of the standard Camera Link cable
Default value.

POCL_LITE

DualBase

MC_SerialLinkB_POCL_LITE
Description The camera-to-frame-grabber serial link is embedded in the Channel Link of PoCL-Lite Camera Link cables

DriverIndex

Base DualBase Full FullXR

Board index in the list of MultiCam compliant boards returned by the driver

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	ADJUST	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
0 << 14	DriverIndex	MC_DriverIndex

Parameter Description

This parameter gives the index of a particular board in the list returned by the driver. This parameter is used to access the Board object parameters related to the board.

The MultiCam compliant boards are assigned consecutive integer numbers starting at 0. The indexing order is system dependent.

PCIPosition

Board index in the list of PCI slots

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	ADJUST	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
1 << 14	PCIPosition	MC_PCIPosition

Parameter Description

This parameter gives the index of the PCI slot associated to a board.

This number is assigned by the operating system in a non-predictable way, but remains consistent for a given configuration in a given system.

BoardName

Name of the board

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	ADJUST	String	Set and Get

Num ID	String Identifier	C, C++ identifier
2 << 14	BoardName	MC_BoardName

Parameter Description

This parameter returns the name of the board. The name is a string of maximum 16 ASCII characters.

BoardIdentifier

Identifier of the board, made by the combination of its type and serial number

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	ADJUST	String	Get Only

Num ID	String Identifier	C, C++ identifier
3 << 14	BoardIdentifier	MC_BoardIdentifier

Parameter Description

This parameter gives the board type and its serial number, providing a unique way to designate a Euresys board.

The board identifier is an ASCII character string, resulting from the concatenation of the board type and the serial number, with an intervening underscore. The serial number is a 6-digit string made of characters 0 to 9;for instance, GRABLINK_FULL_000123.

Refer to BoardType for available board types.

NameBoard

Naming of the selected board

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	ADJUST	String	Set Only

Num ID	String Identifier	C, C++ identifier
4 << 14	NameBoard	MC_NameBoard

Parameter Description

Setting this parameter writes the name to the selected board. This name is stored inside an on-board non-volatile memory.

The name is a string of maximum 16 ASCII characters.

SerialNumber

Base DualBase Full FullXR

Unique serial number of the board

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	ADJUST	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
5 << 14	SerialNumber	MC_SerialNumber

Parameter Description

This parameter returns the serial number assigned to the selected board. This 6-digit number is unique for a board of a given type.

Parameter Values

Value	Description
0	*Minimum range value.*
999999	*Maximum range value.*

BoardType

Type of the board

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	ADJUST	Enumerated	Get Only

Num ID	String Identifier	C, C++ identifier
6 << 14	BoardType	MC_BoardType

Parameter Values

GRABLINK_BASE

Base

MC_BoardType_GRABLINK_BASE

Description
1624 Grablink Base

GRABLINK_DUALBASE

DualBase

MC_BoardType_GRABLINK_DUALBASE

Description
1623 Grablink DualBase

GRABLINK_FULL

Full

MC_BoardType_GRABLINK_FULL

Description
1622 Grablink Full

GRABLINK_FULL_XR

FullXR

MC_BoardType_GRABLINK_FULL_XR

Description
1626 Grablink Full XR

SerialControlA

Base DualBase Full FullXR

Creation of a serial link through a virtual COM port

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	ADJUST	String	Set and Get

Num ID	String Identifier	C, C++ identifier
70 << 14	SerialControlA	MC_SerialControlA

Parameter Description

This parameter declares which virtual COM port is associated with the serial link associated with camera connector M.

This parameter requires the user who runs the application to have administrator privileges. Moreover, under Windows Vista and later, the application must be explicitly run as administrator.

```
Status = McSetParamStr(MC_BOARD+1, MC_SerialControlA, "COM4");
```

SerialControlB

DualBase

Creation of a serial link through virtual COM port

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	ADJUST	String	Set and Get

Num ID	String Identifier	C, C++ identifier
71 << 14	SerialControlB	MC_SerialControlB

Parameter Description

This parameter declares which virtual COM port is associated with the serial link associated with camera connector B.

This parameter requires the user who runs the application to have administrator privileges. Moreover, under Windows Vista and later, the application must be explicitly run as administrator.

```
Status = McSetParamStr(MC_BOARD+1, MC_SerialControlB, "COM5");
```

PCIeDeviceID

Base DualBase Full FullXR

Identification number assigned to the board on the PCI Express system

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	ADJUST	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
2911 << 14	PCIeDeviceID	MC_PCIeDeviceID

Parameter Description

Getting this parameter returns the board ID on the PCI Express system (when the board is configured in normal mode).

Parameter Values

Base DualBase Full FullXR

Value	Description
778	**1622 Grablink Full** - Normal mode
779	**1622 Grablink Full** - Recovery mode
780	**1623 Grablink DualBase** - Normal mode
781	**1623 Grablink DualBase** - Recovery mode
782	**1624 Grablink Base** - Normal mode
783	**1624 Grablink Base** - Recovery mode
784	**1626 Grablink Full XR** - Normal mode
785	**1626 Grablink Full XR** - Recovery mode

PCIeLinkWidth

Base DualBase Full FullXR

Negotiated width of the PCI Express link

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	ADJUST	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
10310 << 14	PCIeLinkWidth	MC_PCIeLinkWidth

Parameter Values

Base

Value	Description
1	1 lane

DualBase Full FullXR

Value	Description
1	1 lane
4	4 lanes

PCIePayloadSize

Base DualBase Full FullXR

Negotiated payload size of the Transport Layer Packets (TLP)

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	ADJUST	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
10403 << 14	PCIePayloadSize	MC_PCIePayloadSize

Parameter Values

Base DualBase Full FullXR

Value	Description
128	128 bytes
256	256 bytes
512	512 bytes
1024	1024 bytes

PCIeEndPointRevisionId

Base DualBase Full FullXR

Revision number of the PCI Express end point firmware

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	ADJUST	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
10311 << 14	PCIeEndPointRevisionId	MC_PCIeEndPointRevisionId

PoCL_PowerInput

Base DualBase FullXR

Status of the camera power input

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	EXPERT	Enumerated	Get Only

Num ID	String Identifier	C, C++ identifier
9943 << 14	PoCL_PowerInput	MC_PoCL_PowerInput

Parameter Values

ON

Base DualBase FullXR

MC_PoCL_PowerInput_ON
Description A 12V power supply is connected to the camera power connector.

OFF

Base DualBase FullXR

MC_PoCL_PowerInput_OFF
Description No power supply is connected to the camera power connector.

OemSafetyLock

Control for locking and checking the board

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	EXPERT	String	Set and Get

Num ID	String Identifier	C, C++ identifier
8 << 14	OemSafetyLock	MC_OemSafetyLock

Parameter Description

This parameter, along with OemSafetyKey , provides a method to assign a safety key to the selected board. The key is an 8-byte string of ASCII characters. Any character is allowed. A null character acts as the termination character of the safety key.

The value when "set" is an 8-byte string of ASCII characters. The entered key is stored in the non-volatile memory of the board and cannot be read back. The "set" operation fails if the key is longer than 8 characters. In that case, the returned error code is MC_INVALID_VALUE.

The value when "get" is the string TRUE or FALSE, that is the validity of the key, which has been previously entered under OemSafetyKey .

See also "Board Security Feature" on page 496.

OemSafetyKey

Safety key for key checking

Parameter Info

Class	Category	Level	Type	Access
Board	Board Information	EXPERT	String	Set Only

Num ID	String Identifier	C, C++ identifier
9 << 14	OemSafetyKey	MC_OemSafetyKey

Parameter Description

This parameter, along with OemSafetyLock , provides a method to assign a safety key to the selected board. The key is implemented as a an 8-byte string of ASCII characters. Any character is allowed. A null character acts as the termination character of the safety key.

The key is stored in the non-volatile memory of the board and cannot be read back.

The validity of the key is returned by OemSafetyLock .

A "set" operation on the OemSafetyLock parameter fails if the key is longer than 8 characters. The returned error code is MC_INVALID_VALUE.

See also "Board Security Feature" on page 496.

3.2. Input/Output Control Category

Parameters providing access to input and output digital lines featured by the board

InputConfig

Base DualBase Full FullXR

Setting of the I/O lines used as inputs

Parameter Info

Class	Category	Level	Type	Access
Board	Input/Output Control	ADJUST	Enumerated collection	Set Only

Num ID	String Identifier	C, C++ identifier
1733 << 14	InputConfig	MC_InputConfig

Parameter Description

The item number of this collection parameter is used as an index to point the relevant input designator among the set of designators owned by selected board.

Refer to "I/O Indices Catalog" on page 490 for a list of I/O indices.

Parameter Values

SOFT

Base DualBase Full FullXR

MC_InputConfig_SOFT
Description Declares that the I/O line is locked for general-purpose software input function.

FREE

Base DualBase Full FullXR

MC_InputConfig_FREE
Description Declares the I/O line to be used for any allowed function.

OutputConfig

Base DualBase Full FullXR

Configuration of the I/O lines used as outputs

Parameter Info

Class	Category	Level	Type	Access
Board	Input/Output Control	ADJUST	Enumerated collection	Set Only

Num ID	String Identifier	C, C++ identifier
1740 << 14	OutputConfig	MC_OutputConfig

Parameter Description

The item number of this collection parameter is used as an index to point the relevant input designator among the set of designators owned by selected board.

Refer to "I/O Indices Catalog" on page 490 for a list of I/O indices.

Parameter Values

SOFT

Base DualBase Full FullXR

MC_OutputConfig_SOFT

Description
Declares that the I/O line is locked for general-purpose software output function.

FREE

Base DualBase Full FullXR

MC_OutputConfig_FREE

Description
Declares the I/O line to be used for any allowed function.

EVENT

Base DualBase Full FullXR

MC_OutputConfig_EVENT

Description
Declares the I/O line to be used to report an event.

InputFunction

Report of the I/O lines used as inputs

Parameter Info

Class	Category	Level	Type	Access
Board	Input/Output Control	ADJUST	Enumerated collection	Get Only

Num ID	String Identifier	C, C++ identifier
1734 << 14	InputFunction	MC_InputFunction

Parameter Description

The item number of this collection parameter is used as an index to point the relevant input designator among the set of designators owned by selected board.

Refer to "I/O Indices Catalog" on page 490 for a list of I/O indices.

The values are specific to each collection member. For further information, refer to the handbooks.

Parameter Values

FREE

Base DualBase Full FullXR

MC_InputFunction_FREE

Description
The input line is free from software and channel use.

Default value.

SOFT

Base DualBase Full FullXR

MC_InputFunction_SOFT

Base DualBase Full FullXR

Description
The I/O line is used as a general-purpose software-controlled input.

NONE

MC_InputFunction_NONE

Description
The I/O line does not exist.

UNKNOWN

MC_InputFunction_UNKNOWN

Description
The functional input usage of the I/O line cannot be determined.

LVAL

Base DualBase Full FullXR

MC_InputFunction_LVAL

Description
The I/O line is used to monitor a channel link LVAL.

FVAL

Base DualBase Full FullXR

MC_InputFunction_FVAL

Description
The I/O line is used to monitor a channel link FVAL.

DVAL

Base DualBase Full FullXR

MC_InputFunction_DVAL

Description
The I/O line is used to monitor a channel link DVAL.

SPARE

Base DualBase Full FullXR

MC_InputFunction_SPARE

Description
The I/O line is used to monitor a channel link SPARE.

CK_PRESENT

Base DualBase Full FullXR

MC_InputFunction_CK_PRESENT

Description
The I/O line is used for channel link clock presence indication.

POWERSTATE5V

Base DualBase Full FullXR

MC_InputFunction_POWERSTATE5V

Description
The I/O line is used for 5V power presence indication.

POWERSTATE12V

Base DualBase Full FullXR

MC_InputFunction_POWERSTATE12V

Description
The I/O line is used for 12V power presence indication.

OutputFunction

Report of the I/O lines used as outputs

Parameter Info

Class	Category	Level	Type	Access
Board	Input/Output Control	ADJUST	Enumerated collection	Get Only

Num ID	String Identifier	C, C++ identifier
1741 << 14	OutputFunction	MC_OutputFunction

Parameter Description

The item number of this collection parameter is used as an index to point the relevant input designator among the set of designators owned by selected board.

Refer to "I/O Indices Catalog" on page 490 for a list of I/O indices.

The values are specific to each collection member. For further information, refer to the handbooks.

Parameter Values

SOFT

Base DualBase Full FullXR

MC_OutputFunction_SOFT

Base DualBase Full FullXR

Description
The I/O line is used as a general-purpose software-controlled output.

FREE

Base DualBase Full FullXR

MC_OutputFunction_FREE

Base DualBase Full FullXR

Description
The I/O line is free from software or channel use.

Default value.

NONE

MC_OutputFunction_NONE

Description
The I/O line does not exist.

UNKNOWN

MC_OutputFunction_UNKNOWN

Description
The functional output usage of the I/O line cannot be determined.

InputState

Report of the logic state of I/O lines used as inputs

Parameter Info

Class	Category	Level	Type	Access
Board	Input/Output Control	ADJUST	Enumerated collection	Get Only

Num ID	String Identifier	C, C++ identifier
1735 << 14	InputState	MC_InputState

Parameter Description

The item number of this collection parameter is used as an index to point the relevant input designator among the set of designators owned by selected board.

Refer to "I/O Indices Catalog" on page 490 for a list of I/O indices.

Getting the InputState enumerated parameter delivers the present status of the interrogated input line.

- The value NONE is reported when the corresponding InputFunction parameter is UNKNOWN.
- A MultiCam error is reported when the corresponding InputFunction parameter is NONE.

Parameter Values

LOW

Base DualBase Full FullXR

MC_InputState_LOW

Description
Presently at the low logic state.

Base DualBase Full FullXR

Description
For isolated current-sense inputs: input current < 1 mA, or unconnected input port
For high-speed differential inputs: input voltage (VIN+ - VIN-) < VThreshold

HIGH

Base DualBase Full FullXR

MC_InputState_HIGH

Description
Presently at the high logic state.

Base DualBase Full FullXR

Description
For isolated current-sense inputs, input current > 1 mA
For high-speed differential inputs, input voltage (VIN+ - VIN-) > VThreshold, or unconnected input port

OutputState

Logic state of I/O lines used as outputs

Parameter Info

Class	Category	Level	Type	Access
Board	Input/Output Control	ADJUST	Enumerated collection	Set and Get

Num ID	String Identifier	C, C++ identifier
1742 << 14	OutputState	MC_OutputState

Parameter Description

The item number of this collection parameter is used as an index to point the relevant input designator among the set of designators owned by selected board.

Refer to "I/O Indices Catalog" on page 490 for a list of I/O indices.

Getting the OutputState parameter is only allowed when the corresponding OutputFunction parameter is SOFT.

The returned value is the one that has been previously set.

The value NONE is reported when the corresponding OutputFunction parameter is other than SOFT.

Parameter Values

LOW

Base DualBase Full FullXR

MC_OutputState_LOW

Base DualBase Full FullXR

Description
The contact switch of isolated outputs is open(OFF).
Initial state after Power-On.

HIGH

Base DualBase Full FullXR

MC_OutputState_HIGH

Base DualBase Full FullXR

Description
The contact switch is closed (ON).

TOGGLE

Base DualBase Full FullXR

MC_OutputState_TOGGLE

Description
A logic state opposite to the present one is issued.

NONE

Base DualBase Full FullXR

MC_OutputState_NONE

Description
The I/O line is not presently used as an output.

SetSignal

Base DualBase Full FullXR

Event source selection to set the EVENT register

Parameter Info

Class	Category	Level	Type	Access
Board	Input/Output Control	ADJUST	Enumerated collection	Set and Get

Num ID	String Identifier	C, C++ identifier
10575 << 14	SetSignal	MC_SetSignal

Parameter Description

Selects an event source to set the EVENT register driving the EVENT signal of the selected output port.

Parameter Usage

Relevance condition(s):

Condition: OutputConfig is set to EVENT.

Parameter Values

NONE

Base DualBase Full FullXR

MC_SetSignal_NONE
Description All event sources are disconnected.
Default value.

SCA

Base DualBase Full FullXR

MC_SetSignal_SCA
Description The 'Start Channel Activity' event source is selected.

ECA

Base DualBase Full FullXR

MC_SetSignal_ECA

Description
The 'End Channel Activity' event source is selected.

SAP

Base DualBase Full FullXR

MC_SetSignal_SAP

Description
The 'Start Acquisition Phase' event source is selected.

EAP

Base DualBase Full FullXR

MC_SetSignal_EAP

Description
The 'End Acquisition Phase' event source is selected.

SAS

Base DualBase Full FullXR

MC_SetSignal_SAS

Description
The 'Start Acquisition Sequence' event source is selected.

EAS

Base DualBase Full FullXR

MC_SetSignal_EAS

Description
The 'End Acquisition Sequence' event source is selected.

FVAL_GOHIGH

Base DualBase Full FullXR

MC_SetSignal_FVAL_GOHIGH

Description
The 'FVAL Going High' event source is selected.

FVAL_GOLOW

Base DualBase Full FullXR

MC_SetSignal_FVAL_GOLOW

Description
The 'FVAL Going Low' event source is selected.

LVAL_GOHIGH

Base DualBase Full FullXR

MC_SetSignal_LVAL_GOHIGH

Description
The 'LVAL Going High' event source is selected.

LVAL_GOLOW

Base DualBase Full FullXR

MC_SetSignal_LVAL_GOLOW

Description
The 'LVAL Going Low' event source is selected.

DVAL_GOHIGH

Base DualBase Full FullXR

MC_SetSignal_DVAL_GOHIGH

Description
The 'DVAL Going High' event source is selected.

DVAL_GOLOW

Base DualBase Full FullXR

MC_SetSignal_DVAL_GOLOW

Description
The 'DVAL Going Low' event source is selected.

CC1_GOHIGH

Base DualBase Full FullXR

MC_SetSignal_CC1_GOHIGH

Description
The 'CC1 Going High' event source is selected.

CC1_GOLOW

Base DualBase Full FullXR

MC_SetSignal_CC1_GOLOW

Description
The 'CC1 Going Low' event source is selected.

CC2_GOHIGH

Base DualBase Full FullXR

MC_SetSignal_CC2_GOHIGH

Description
The 'CC2 Going High' event source is selected.

CC2_GOLOW

Base DualBase Full FullXR

MC_SetSignal_CC2_GOLOW

Description
The 'CC2 Going Low' event source is selected.

CC3_GOHIGH

Base DualBase Full FullXR

MC_SetSignal_CC3_GOHIGH

Description
The 'CC3 Going High' event source is selected.

CC3_GOLOW

Base DualBase Full FullXR

MC_SetSignal_CC3_GOLOW

Description
The 'CC3 Going Low' event source is selected.

CC4_GOHIGH

Base DualBase Full FullXR

MC_SetSignal_CC4_GOHIGH

Description
The 'CC4 Going High' event source is selected.

CC4_GOLOW

Base | DualBase | Full | FullXR

MC_SetSignal_CC4_GOLOW

Description
The 'CC4 Going Low' event source is selected.

ResetSignal

Base DualBase Full FullXR

Event source selection to reset the EVENT register

Parameter Info

Class	Category	Level	Type	Access
Board	Input/Output Control	ADJUST	Enumerated collection	Set and Get

Num ID	String Identifier	C, C++ identifier
10576 << 14	ResetSignal	MC_ResetSignal

Parameter Description

Selects an event source to reset the EVENT register driving the EVENT signal of the selected output port.

Parameter Usage

Relevance condition(s):

Condition: OutputConfig is set to EVENT.

Parameter Values

NONE

Base DualBase Full FullXR

MC_ResetSignal_NONE
Description All event sources are disconnected.
Default value.

SCA

Base DualBase Full FullXR

MC_ResetSignal_SCA
Description The 'Start Channel Activity' event source is selected.

ECA

Base DualBase Full FullXR

MC_ResetSignal_ECA

Description
The 'End Channel Activity' event source is selected.

SAP

Base DualBase Full FullXR

MC_ResetSignal_SAP

Description
The 'Start Acquisition Phase' event source is selected.

EAP

Base DualBase Full FullXR

MC_ResetSignal_EAP

Description
The 'End Acquisition Phase' event source is selected.

SAS

Base DualBase Full FullXR

MC_ResetSignal_SAS

Description
The 'Start Acquisition Sequence' event source is selected.

EAS

Base DualBase Full FullXR

MC_ResetSignal_EAS

Description
The 'End Acquisition Sequence' event source is selected.

FVAL_GOHIGH

Base DualBase Full FullXR

MC_ResetSignal_FVAL_GOHIGH

Description
The 'FVAL Going High' event source is selected.

FVAL_GOLOW

Base DualBase Full FullXR

MC_ResetSignal_FVAL_GOLOW

Description
The 'FVAL Going Low' event source is selected.

LVAL_GOHIGH

Base DualBase Full FullXR

MC_ResetSignal_LVAL_GOHIGH

Description
The 'LVAL Going High' event source is selected.

LVAL_GOLOW

Base DualBase Full FullXR

MC_ResetSignal_LVAL_GOLOW

Description
The 'LVAL Going Low' event source is selected.

DVAL_GOHIGH

Base DualBase Full FullXR

MC_ResetSignal_DVAL_GOHIGH

Description
The 'DVAL Going High' event source is selected.

DVAL_GOLOW

Base DualBase Full FullXR

MC_ResetSignal_DVAL_GOLOW

Description
The 'DVAL Going Low' event source is selected.

CC1_GOHIGH

Base DualBase Full FullXR

MC_ResetSignal_CC1_GOHIGH

Description
The 'CC1 Going High' event source is selected.

CC1_GOLOW

Base DualBase Full FullXR

MC_ResetSignal_CC1_GOLOW

Description
The 'CC1 Going Low' event source is selected.

CC2_GOHIGH

Base DualBase Full FullXR

MC_ResetSignal_CC2_GOHIGH

Description
The 'CC2 Going High' event source is selected.

CC2_GOLOW

Base DualBase Full FullXR

MC_ResetSignal_CC2_GOLOW

Description
The 'CC2 Going Low' event source is selected.

CC3_GOHIGH

Base DualBase Full FullXR

MC_ResetSignal_CC3_GOHIGH

Description
The 'CC3 Going High' event source is selected.

CC3_GOLOW

Base DualBase Full FullXR

MC_ResetSignal_CC3_GOLOW

Description
The 'CC3 Going Low' event source is selected.

CC4_GOHIGH

Base DualBase Full FullXR

MC_ResetSignal_CC4_GOHIGH

Description
The 'CC4 Going High' event source is selected.

CC4_GOLOW

Base | DualBase | Full | FullXR

MC_ResetSignal_CC4_GOLOW

Description
The 'CC4 Going Low' event source is selected.

InputStyle

Electrical style of I/O lines used as inputs

Parameter Info

Class	Category	Level	Type	Access
Board	Input/Output Control	EXPERT	Enumerated collection	Set and Get

Num ID	String Identifier	C, C++ identifier
1736 << 14	InputStyle	MC_InputStyle

Parameter Description

The item number of this collection parameter is used as an index to point the relevant input designator among the set of designators owned by selected board.

Refer to "I/O Indices Catalog" on page 490 for a list of I/O indices.

Setting InputStyle to a precise value yields better electrical performance, such as better common mode rejection ratio.

The values are specific to each collection member. For further information, refer to the handbooks.

Parameter Values

CHANNELLINK

Base DualBase Full FullXR

MC_InputStyle_CHANNELLINK
Description The input line is a signal embedded in Channel Link.

ISO

Base DualBase Full FullXR

MC_InputStyle_ISO
Description The input line is an isolated current-sense input with wide voltage input range up to 30V, compatible with totem-pole LVTTL, TTL, 5V CMOS drivers, RS-422 differential line drivers, potential free contacts, solid-state relays and opto-couplers.

DIFF

Base DualBase Full FullXR

MC_InputStyle_DIFF

Description
The input line is a high-speed differential input compatible with ANSI/EIA/TIA-422/485 differential line drivers and complementary TTL drivers.

POWERSTATE

Base DualBase Full FullXR

MC_InputStyle_POWERSTATE

Description
The input line reports the state of a power input.

OutputStyle

Electrical style of I/O lines used as outputs

Parameter Info

Class	Category	Level	Type	Access
Board	Input/Output Control	EXPERT	Enumerated collection	Set and Get

Num ID	String Identifier	C, C++ identifier
1748 << 14	OutputStyle	MC_OutputStyle

Parameter Description

The item number of this collection parameter is used as an index to point the relevant input designator among the set of designators owned by selected board.

Refer to "I/O Indices Catalog" on page 490 for a list of I/O indices.

The values are specific to each collection member. For further information, refer to the handbooks.

Parameter Values

OPTO

Base DualBase Full FullXR

MC_OutputStyle_OPTO

Base DualBase Full FullXR

Description
Isolated contact outputs compatible with 30V / 100mA loads.

LVDS

Base DualBase Full FullXR

MC_OutputStyle_LVDS

Description
The output line is differential LVDS, RS-422 or RS-485 compatible.

InputPinName

Pin name of the I/O line used as input

Parameter Info

Class	Category	Level	Type	Access
Board	Input/Output Control	EXPERT	Enumerated collection	Get Only

Num ID	String Identifier	C, C++ identifier
1796 << 14	InputPinName	MC_InputPinName

Parameter Description

The item number of this collection parameter is used as an index to point the relevant input designator among the set of designators owned by selected board.

Refer to "I/O Indices Catalog" on page 490 for a list of I/O indices.

The values are specific to each collection member. For further information, refer to the handbooks.

Parameter Values

UNKNOWN

MC_InputPinName_UNKNOWN

Description
The I/O line does not exist.

FVAL

Base DualBase

MC_InputPinName_FVAL

Description
The I/O line is issued from Camera Link camera connector pins named FVAL.

DVAL

Base DualBase

MC_InputPinName_DVAL

Description
The I/O line is issued from Camera Link camera connector pins named DVAL.

LVAL

Base DualBase

MC_InputPinName_LVAL

Description
The I/O line is issued from Camera Link camera connector pins named LVAL.

SPARE

Base DualBase

MC_InputPinName_SPARE

Description
The I/O line is issued from Camera Link camera connector pins named SPARE.

IIN1

Base DualBase Full FullXR

MC_InputPinName_IIN1

Description
The I/O line is issued from an I/O connector pin named IIN1.

IIN2

Base DualBase Full FullXR

MC_InputPinName_IIN2

Description
The I/O line is issued from an I/O connector pin named IIN2.

IIN3

Base DualBase Full FullXR

MC_InputPinName_IIN3

Description
The I/O line is issued from an I/O connector pin named IIN3.

IIN4

Base DualBase Full FullXR

MC_InputPinName_IIN4

Description
The I/O line is issued from an I/O connector pin named IIN4.

DIN1

Base DualBase Full FullXR

MC_InputPinName_DIN1

Description
The I/O line is issued from an I/O connector pin named DIN1.

DIN2

Base DualBase Full FullXR

MC_InputPinName_DIN2

Description
The I/O line is issued from an I/O connector pin named DIN2.

LVAL_X

Full FullXR

MC_InputPinName_LVAL_X

Description
The I/O line is issued from Camera Link Channel X camera connector pins named LVAL.

FVAL_X

Full FullXR

MC_InputPinName_FVAL_X

Description
The I/O line is issued from Camera Link Channel X camera connector pins named FVAL.

DVAL_X

Full FullXR

MC_InputPinName_DVAL_X

Description
The I/O line is issued from Camera Link Channel X camera connector pins named DVAL.

SPARE_X

Full FullXR

MC_InputPinName_SPARE_X

Description
The I/O line is issued from Camera Link Channel X camera connector pins named SPARE.

CK_PRESENT_X

Full FullXR

MC_InputPinName_CK_PRESENT_X

Description
The I/O line is issued from the Camera Link Channel X clock presence detector.

LVAL_Y

Full FullXR

MC_InputPinName_LVAL_Y

Description
The I/O line is issued from Camera Link Channel Y camera connector pins named LVAL.

FVAL_Y

Full FullXR

MC_InputPinName_FVAL_Y

Description
The I/O line is issued from Camera Link Channel Y camera connector pins named FVAL.

DVAL_Y

Full FullXR

MC_InputPinName_DVAL_Y

Description
The I/O line is issued from Camera Link Channel Y camera connector pins named DVAL.

SPARE_Y

Full FullXR

MC_InputPinName_SPARE_Y

Description
The I/O line is issued from Camera Link Channel Y camera connector pins named SPARE.

CK_PRESENT_Y

Full FullXR

MC_InputPinName_CK_PRESENT_Y

Description
The I/O line is issued from the Camera Link Channel Y clock presence detector.

LVAL_Z

Full FullXR

MC_InputPinName_LVAL_Z

Description
The I/O line is issued from Camera Link Channel Z camera connector pins named LVAL.

FVAL_Z

Full FullXR

MC_InputPinName_FVAL_Z

Description
The I/O line is issued from Camera Link Channel Z camera connector pins named FVAL.

DVAL_Z

Full FullXR

MC_InputPinName_DVAL_Z

Description
The I/O line is issued from Camera Link Channel Z camera connector pins named DVAL.

SPARE_Z

Full FullXR

MC_InputPinName_SPARE_Z

Description
The I/O line is issued from Camera Link Channel Z camera connector pins named SPARE.

CK_PRESENT_Z

Full FullXR

MC_InputPinName_CK_PRESENT_Z

Description
The I/O line is issued from the Camera Link Channel Z clock presence detector.

POWER_5V

Base DualBase Full FullXR

MC_InputPinName_POWER_5V

Description
The I/O line is issued from the voltage monitor of the +5 V power input.

POWER_12V

Base DualBase Full FullXR

MC_InputPinName_POWER_12V

Description
The I/O line is issued from the voltage monitor of the +12 V power input.

OutputPinName

Pin name of the I/O line used as the output

Parameter Info

Class	Category	Level	Type	Access
Board	Input/Output Control	EXPERT	Enumerated collection	Get Only

Num ID	String Identifier	C, C++ identifier
1798 << 14	OutputPinName	MC_OutputPinName

Parameter Description

The item number of this collection parameter is used as an index to point the relevant output designator among the set of designators owned by selected board.

Refer to "I/O Indices Catalog" on page 490 for a list of I/O indices.

The values are specific to each collection member. For further information, refer to the handbooks.

Parameter Values

UNKNOWN

Full

MC_OutputPinName_UNKNOWN

Description
The I/O line does not exist.

CC1

Base DualBase Full FullXR

MC_OutputPinName_CC1

Description
The I/O line is driving Camera Link connector pin named CC1.

CC2

Base DualBase Full FullXR

MC_OutputPinName_CC2

Description
The I/O line is driving Camera Link connector pin named CC2.

CC3

Base DualBase Full FullXR

MC_OutputPinName_CC3

Description
The I/O line is driving Camera Link connector pin named CC3.

CC4

Base DualBase Full FullXR

MC_OutputPinName_CC4

Description
The I/O line is driving Camera Link connector pin named CC4.

IOUT1

Base DualBase Full FullXR

MC_OutputPinName_IOUT1

Description
The I/O line is driving connector pin named IOUT1.

IOUT2

Base DualBase Full FullXR

MC_OutputPinName_IOUT2

Description
The I/O line is driving connector pin named IOUT2.

IOUT3

Base DualBase Full FullXR

MC_OutputPinName_IOUT3

Description
The I/O line is driving connector pin named IOUT3.

IOUT4

Base DualBase Full FullXR

MC_OutputPinName_IOUT4

Description
The I/O line is driving connector pin named IOUT4.

ConnectorName

Connector name of the I/O lines used as input

Parameter Info

Class	Category	Level	Type	Access
Board	Input/Output Control	EXPERT	Enumerated collection	Get Only

Num ID	String Identifier	C, C++ identifier
1815 << 14	ConnectorName	MC_ConnectorName

Parameter Description

The item number of this collection parameter is used as an index to point the relevant input designator among the set of designators owned by selected board.

Refer to "I/O Indices Catalog" on page 490 for a list of I/O indices.

Parameter Values

UNKNOWN

MC_ConnectorName_UNKNOWN

Description
The I/O line does not exist.

IO

Base Full FullXR

MC_ConnectorName_IO

Base Full FullXR

Description
The I/O lines are available on the connector named I/O.

CAMERA

Base Full FullXR

MC_ConnectorName_CAMERA

Description
The I/O lines are available on the connector named Camera.

CAMERA_B

DualBase

MC_ConnectorName_CAMERA_B

Description
The I/O lines are available on the connector named Camera B.

IO_A

DualBase

MC_ConnectorName_IO_A

Description
The I/O lines are available on the connector named I/O A.

IO_B

DualBase

MC_ConnectorName_IO_B

Description
The I/O lines are available on the connector named I/O B.

4. Channel Class

What Is a Channel?

The Channel class groups all MultiCam parameters dedicated to the control of image acquisition related features.

A Channel object is an instance of the Channel class, represented by a dedicated set of such parameters.

Typically, the following items are defined and controlled by the Channel object:

- The camera feeding the channel, including reset and exposure control
- The connector and cable linking the camera to the frame grabber
- The switching structures routing the analog or digital video signal inside the frame grabbe
- In case of analog camera, the analog-to-digital converter and the associated signal conditioning devices
- In case of digital camera, the digital receiving or de-serializing devices
- The timing generator and controller associated to the camera, and the video signal conditioning
- All digital devices affecting the signal during acquisition, performing tasks such as lookup tables, byte alignment, data channel merging...
- The data buffer receiving the images
- The DMA devices extracting images out of the data buffer for transfer into host memory
- The destination cluster of host memory surfaces
- The hardware resources managing the external system trigger

The channel is the association of an individual grabber connected to a camera delivering data to a set of surfaces, called a cluster. The channel is able to transport an image from the camera towards a surface belonging to the cluster and usually located in the host memory.

4.1. Camera Specification Category

Parameters specifying the type and operational mode of the camera feeding the channel

CamFile

Name of the CAM file

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Specification	SELECT	String	Set and Get

Num ID	String Identifier	C, C++ identifier
11 << 14	CamFile	MC_CamFile

Parameter Description

This parameter specifies a camera configuration file as a character string. The .cam extension may or may not be included. The maximum string length is 1024.

Getting this parameter returns the name of the lastly executed CAM file.

Refer to CAM Files for CAM file syntax and location.

See also "CAM Files" on page 515 in the MultiCam user guide for more information.

Camera

Camera model attached to the grabber

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Specification	SELECT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
700 << 14	Camera	MC_Camera

Parameter Description

Together with CamConfig, this parameter defines a coherent set of camera properties.

Parameter Values

MyCameraLink

Base DualBase Full FullXR

MC_Camera_MyCameraLink
Description Generic Camera Link camera

CamConfig

Configuration of the camera model attached to the grabber

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Specification	SELECT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
701 << 14	CamConfig	MC_CamConfig

Parameter Description

Together with Camera, this parameter defines a coherent set of camera properties.

For Grablink products, the parameter complies to the following syntax: <Imaging>xx [xx]<CamMode><Exp>where:

- **Imaging** designates the type of imaging device:
  - L: Line-scan imaging device
  - P: Progressive area-scan imaging device
- **CamMode** designates the main camera operating mode:
  - R: Asynchronous Reset operating mode. The camera initiates an exposure/readout sequence when it gets a "Reset" signal from the frame grabber
  - S: Synchronous operating mode. The camera is free-running and delivers permanently video data
- **Exp** designates the exposure control method:
  - C: The exposure is controlled by the camera
  - G: The exposure is controlled by the frame grabber
  - P: The camera sensor has no exposure control .It is exposed permanently.

Parameter Values

LxxxxRC

Base DualBase Full FullXR

MC_CamConfig_LxxxxRC

Description
Grabber-controlled rate, camera-controlled exposure time, line-scan camera. The exposure duration is set through camera switches or serial control. The camera cycles are triggered by a pulse over a "Reset" line issued by the frame grabber.

Applicability condition(s)
Base DualBase Full FullXR
Condition: Camera is set to MyCameraLink

LxxxxRG

Base DualBase Full FullXR

MC_CamConfig_LxxxxRG

Description
Grabber-controlled line rate, grabber-controlled exposure, line-scan camera. The exposure duration is defined as the active duration of a pulse over a "Reset" line issued by the frame grabber.

Applicability condition(s)
Base DualBase Full FullXR
Condition: Camera is set to MyCameraLink

LxxxxRP

Base DualBase Full FullXR

MC_CamConfig_LxxxxRP

Description
Grabber-controlled rate, permanent exposure, line-scan camera. The camera has no exposure control capability, resulting in permanent exposure. The camera cycles are triggered by a pulse over a "Reset" line issued by the frame grabber.

Applicability condition(s)
Base DualBase Full FullXR
Condition: Camera is set to MyCameraLink

LxxxxSC

Base DualBase Full FullXR

MC_CamConfig_LxxxxSC

Description
Free-running, camera-controlled exposure time, line-scan camera. The exposure duration is set through camera switches or serial control. The camera cycles are free-running.

Applicability condition(s)
Base DualBase Full FullXR
Condition: Camera is set to MyCameraLink

LxxxxSP

Base DualBase Full FullXR

MC_CamConfig_LxxxxSP

Description
Free-running, permanent exposure, line-scan camera. The camera has no exposure control capability, resulting in permanent exposure. The camera cycles are free-running.

Applicability condition(s)
Base DualBase Full FullXR
Condition: Camera is set to MyCameraLink

PxxRC

Base DualBase Full FullXR

MC_CamConfig_PxxRC

Description
Progressive, asynchronous reset operation, camera-controlled exposure, area-scan camera. The exposure duration is set through camera switches or serial control. The camera cycles are triggered by a pulse over a "Reset" line issued by the frame grabber.

Applicability condition(s)
Base DualBase Full FullXR
Condition: Camera is set to MyCameraLink

PxxRG

Base DualBase Full FullXR

MC_CamConfig_PxxRG

Description
Progressive asynchronous reset operation, grabber-controlled exposure, area-scan camera. The exposure duration is defined as the active duration of a pulse over a "Reset" line issued by the frame grabber.

Applicability condition(s)
Base DualBase Full FullXR
Condition: Camera is set to MyCameraLink

PxxSC

Base DualBase Full FullXR

MC_CamConfig_PxxSC

Description

Progressive-scan, synchronous operation, camera-controlled exposure, area-scan camera. The exposure duration is set through camera switches or serial control. The camera cycles are free-running.

Applicability condition(s)

Base DualBase Full FullXR

Condition: Camera is set to MyCameraLink

Imaging

Camera imaging basic geometry

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Specification	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1007 << 14	Imaging	MC_Imaging

Parameter Description

This parameter is used to distinguish the basic kind of camera feeding the channel. See also "Camera Imaging Basic Geometry" on page 501.

Parameter Values

AREA

Base DualBase Full

MC_Imaging_AREA

Description
The currently selected camera is an area-scan model.

LINE

Base DualBase Full FullXR

MC_Imaging_LINE

Description
The currently selected camera is a line-scan model.

TDI

Base DualBase Full FullXR

MC_Imaging_TDI

Description
The currently selected camera is a TDI line-scan model.

Spectrum

Iimaging spectral sensitivity of the specified camera

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Specification	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1008 << 14	Spectrum	MC_Spectrum

Parameter Description

This parameter is used to distinguish the basic kind of camera feeding the channel.

This information only makes sense for frame grabber able to indifferently interface to color or monochrome cameras. See also "Camera Spectral Sensitivity" on page 502.

The way the color information is built at the camera's sensor is further described by the ColorMethod parameter belonging to the "Camera Features Category " on page 108.

Before assigning a value to this parameter, it is mandatory to set Camera and CamConfig .

Parameter Values

BW

Base DualBase Full FullXR

MC_Spectrum_BW
Description The selected camera delivers a monochrome image.

COLOR

Base DualBase Full FullXR

MC_Spectrum_COLOR
Description The selected camera delivers a color image.

IR

Base DualBase Full FullXR

MC_Spectrum_IR
Description The selected camera delivers a monochrome image issued by an infra-red sensor.

DataLink

Data transfer method of the current camera

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Specification	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1009 << 14	DataLink	MC_DataLink

Parameter Description

This parameter is used to return some information on the basic connection structure of the camera feeding the channel. See also "Camera Data Transfer Method" on page 500.

Parameter Values

CAMERALINK

Base DualBase Full FullXR

MC_DataLink_CAMERALINK
Description The camera delivers a digital video signal complying with the Camera Link standard.

4.2. Camera Timing Category

Parameters setting the video timing attributes of the camera feeding the channel

PixelClkMode

Base DualBase Full FullXR

Camera Link clock signal characteristics

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Timing	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10574 << 14	PixelClkMode	MC_PixelClkMode

Parameter Description

Defines how the camera delivers the Camera Link clock signal.

Parameter Usage

Directive: Set to INTERMITTENT when the camera doesn't permanently deliver the Camera Link clock.

Parameter Values

PERMANENT

Base DualBase Full FullXR

MC_PixelClkMode_PERMANENT
Description The camera delivers permanently the Camera Link clock signal
Default value.

INTERMITTENT

Base DualBase Full FullXR

MC_PixelClkMode_INTERMITTENT
Description The camera delivers intermittently the Camera Link clock signal

LineRate_Hz

Camera line repetition rate, expressed in Hertz

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Timing	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
705 << 14	LineRate_Hz	MC_LineRate_Hz

Parameter Description

This parameter declares the line rate, which is the repetition frequency of the video lines scanned and delivered by the camera feeding the channel.

This value is a performance figure often stated as is by the camera manufacturer.

For area-scan cameras, the line rate is usually under control of the camera itself.

In the special case of an area-scan camera receiving horizontal drive information from the frame grabber, the LineRate_Hz parameter expresses the recommended horizontal frequency to be applied to the camera.

For line-scan cameras, the line rate is usually under control of the frame grabber. In that case, the LineRate_Hz parameter declares the maximum line frequency the camera can accept.

In the special case of a line-scan camera controlling its own line timing, the LineRate_Hz parameter expresses the actual horizontal frequency set by the camera.

Parameter Values

Base DualBase Full FullXR

Value	Description
10	10 Hz *Minimum range value.*
100000	100,000 Hz (= 100 kHz) *Maximum range value.*

FrameRate_mHz

Base DualBase Full FullXR

Camera frame repetition rate, expressed in milliHertz

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Timing	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
2222 << 14	FrameRate_mHz	MC_FrameRate_mHz

Parameter Description

This parameter declares the frame rate, which is the repetition frequency of the video frames scanned and delivered by the camera feeding the channel.

This value is a performance figure often stated as is by the camera manufacturer.

Parameter Values

Base DualBase Full FullXR

Value	Description
1000	1,000 milliHertz (=1 Hz) *Minimum range value.*
127500000	127,500,000 milliHertz (=127,5 kHz) *Maximum range value.*

LineDur_ns

Total duration of the video line, expressed in nanoseconds

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Timing	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
732 << 14	LineDur_ns	MC_LineDur_ns

Parameter Description

The total duration of the video line is the inverse value of the camera line repetition rate declared by the LineRate_Hz parameter.

For area-scan cameras, the line duration is the sum of the horizontal blanking period and the active part of the video line. This is a feature of the video standard the camera may comply to.

For line-scan cameras, the line duration is the minimum time to scan a single line. It is a practical way to characterize the top performance of the camera.

Parameter Values

Base DualBase Full FullXR

Value	Description
10000	10,000 nanoseconds (=10 microseconds) *Minimum range value.*
100000000	100,000,000 nanoseconds (=100 milliseconds) *Maximum range value.*

Vactive_Ln

Number of active video lines in the frame

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Timing	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
710 << 14	Vactive_Ln	MC_Vactive_Ln

Parameter Description

An active line is, by definition, a video line where useful visual information can appear. Blanking lines take no part in the count of active lines.

In case of interlaced scanning, Vactive_Ln represents the number of active lines for both fields altogether. This is equivalent to the number of active half-lines per field.

In some cases of dual-tap structure, Vactive_Ln represents the number of active lines for both channels altogether.

This parameter is a measure of the height of the camera active window.

It is used to characterize area-scan cameras. It is meaningless for line-scan cameras.

Parameter Values

Value	Description
1	1 line *Minimum range value.*
65535	65,535 lines *Maximum range value.*

FrameDur_us

Base DualBase Full FullXR

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Timing	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
2223 << 14	FrameDur_us	MC_FrameDur_us

Parameter Description

This parameter is expressed in microseconds.

Parameter Values

Base DualBase Full FullXR

Value	Description
10000	10,000 microseconds (=10 milliseconds) *Minimum range value.*
100000000	100,000,000 microseconds (=100 seconds) *Maximum range value.*

Hactive_Px

Base DualBase Full FullXR

Number of active pixels in the line, expressed as a number of camera sensor pixels

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Timing	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
1021 << 14	Hactive_Px	MC_Hactive_Px

Parameter Description

This parameter is used to characterize digital line-scan or area-scan cameras. It announces the number of horizontal pixels belonging to the sensor that are effectively available at the camera output. This is a measure of the width of the camera active window.

The allowed values are depending on several factors: board type, tap configuration and tap geometry. Refer to the Grablink User Guide for an extensive description of all cases.

VsyncAft_Ln

Base DualBase Full FullXR

Vertical delay between vertical synchronization pulse and camera active window

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Timing	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
712 << 14	VsyncAft_Ln	MC_VsyncAft_Ln

Parameter Description

The delay is expressed as the number of LVAL leading edges to ignore after the leading edge of the FVAL pulse.

Parameter Values

Base DualBase Full FullXR

Value	Description
0	*Minimum range value. Default value.*
255	255 lines after FVAL *Maximum range value.*

HsyncAft_Tk

Base DualBase Full FullXR

Horizontal delay between horizontal synchronization pulse and camera active window, expressed in TCU (Timing Clock Unit) from the Camera Link clock

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Timing	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
1111 << 14	HsyncAft_Tk	MC_HsyncAft_Tk

Parameter Description

This parameter applies to Camera Link compliant digital area-scan cameras that deliver a LVAL horizontal synchronization pulse used by the frame grabber to monitor the camera timing.

For line-scan cameras, the delay is counted from the leading edge of LVAL delivered by the camera to the beginning of the read-out period.

Parameter Values

Base DualBase Full FullXR

Value	Description
-1	1 Camera Link clock before LVAL *Minimum range value.*
1023	1023 Camera Link clocks after LVAL *Maximum range value.*

ExposeRecovery_us

Base DualBase Full FullXR

Minimum delay between successive expose pulses, expressed in microseconds

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Timing	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
1311 << 14	ExposeRecovery_us	MC_ExposeRecovery_us

Parameter Description

This parameter declares the minimum amount of time required by the camera between successive expose pulses. Its value is strictly positive.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 microsecond *Minimum range value.*
1000000	1,000,000 microseconds (=1 second) *Maximum range value.*

ReadoutRecovery_us

Base DualBase Full FullXR

Minimum delay between successive read-out phases, expressed in microseconds

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Timing	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
1312 << 14	ReadoutRecovery_us	MC_ReadoutRecovery_us

Parameter Description

This parameter declares the minimum amount of time required by the camera between successive read-out phases.

This is applicable to area-scan cameras only.

The value is strictly positive.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 microsecond *Minimum range value.*
1000000	1,000,000 microseconds (=1 second) *Maximum range value.*

4.3. Camera Features Category

Parameters setting the hardware interface attributes of the camera feeding the channel

TapConfiguration

Base DualBase Full FullXR

Camera Link tap configuration

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
4268 << 14	TapConfiguration	MC_TapConfiguration

Parameter Description

This parameter declares the Camera Link tap configuration used by the camera.

Refer to "TapConfiguration Glossary" on page 485 for terms definitions and naming conventions.

Parameter Values

BASE_1T8

Base DualBase Full FullXR

MC_TapConfiguration_BASE_1T8
Description The camera requires the Camera Link Base configuration to deliver 1 8-bit pixel every clock cycle.

BASE_1T10

Base DualBase Full FullXR

MC_TapConfiguration_BASE_1T10
Description The camera requires the Camera Link Base configuration to deliver 1 10-bit pixel every clock cycle.

BASE_1T12

Base DualBase Full FullXR

MC_TapConfiguration_BASE_1T12

Description
The camera requires the Camera Link Base configuration to deliver 1 12-bit pixel every clock cycle.

BASE_1T14

Base DualBase Full FullXR

MC_TapConfiguration_BASE_1T14

Description
The camera requires the Camera Link Base configuration to deliver 1 14-bit pixel every clock cycle.

BASE_1T16

Base DualBase Full FullXR

MC_TapConfiguration_BASE_1T16

Description
The camera requires the Camera Link Base configuration to deliver 1 16-bit pixel every clock cycle.

BASE_1T24

Base DualBase Full FullXR

MC_TapConfiguration_BASE_1T24

Description
The camera requires the Camera Link Base configuration to deliver 3 8-bit color components for 1 24-bit pixel every clock cycle.

BASE_2T8

Base DualBase Full FullXR

MC_TapConfiguration_BASE_2T8

Description
The camera requires the Camera Link Base configuration to deliver 2 8-bit pixels every clock cycle.

BASE_2T10

Base DualBase Full FullXR

MC_TapConfiguration_BASE_2T10

Description
The camera requires the Camera Link Base configuration to deliver 2 10-bit pixels every clock cycle.

BASE_2T12

Base DualBase Full FullXR

MC_TapConfiguration_BASE_2T12

Description
The camera requires the Camera Link Base configuration to deliver 2 12-bit pixels every clock cycle.

BASE_3T8

Base DualBase Full FullXR

MC_TapConfiguration_BASE_3T8

Description
The camera requires the Camera Link Base configuration to deliver 3 8-bit pixels every clock cycle.

MEDIUM_1T30

Full FullXR

MC_TapConfiguration_MEDIUM_1T30

Description
The camera requires the Camera Link Medium configuration to deliver 3 10-bit color components for 1 30-bit pixel every clock cycle.

MEDIUM_1T36

Full FullXR

MC_TapConfiguration_MEDIUM_1T36

Description
The camera requires the Camera Link Medium configuration to deliver 3 12-bit color components for 1 36-bit pixel every clock cycle.

MEDIUM_1T42

Full FullXR

MC_TapConfiguration_MEDIUM_1T42

Description
The camera requires the Camera Link Medium configuration to deliver 3 14-bit color components for 1 42-bit pixel every clock cycle.

MEDIUM_1T48

Full FullXR

MC_TapConfiguration_MEDIUM_1T48

Description
The camera requires the Camera Link Medium configuration to deliver 3 16-bit color components for 1 48-bit pixel every clock cycle.

MEDIUM_2T14

Full FullXR

MC_TapConfiguration_MEDIUM_2T14

Description
The camera requires the Camera Link Medium configuration to deliver 2 14-bit pixels every clock cycle.

MEDIUM_2T16

Full FullXR

MC_TapConfiguration_MEDIUM_2T16

Description
The camera requires the Camera Link Medium configuration to deliver 2 16-bit pixels every clock cycle.

MEDIUM_2T24

Full FullXR

MC_TapConfiguration_MEDIUM_2T24

Description
The camera requires the Camera Link Medium configuration to deliver 6 8-bit color components for 2 24-bit pixels every clock cycle.

MEDIUM_3T10

Full FullXR

MC_TapConfiguration_MEDIUM_3T10

Description
The camera requires the Camera Link Medium configuration to deliver 3 10-bit pixels every clock cycle.

MEDIUM_3T12

Full FullXR

MC_TapConfiguration_MEDIUM_3T12

Description
The camera requires the Camera Link Medium configuration to deliver 3 12-bit pixels every clock cycle.

MEDIUM_3T14

Full FullXR

MC_TapConfiguration_MEDIUM_3T14

Description
The camera requires the Camera Link Medium configuration to deliver 3 14-bit pixels every clock cycle.

MEDIUM_3T16

Full FullXR

MC_TapConfiguration_MEDIUM_3T16

Description
The camera requires the Camera Link Medium configuration to deliver 3 16-bit pixels every clock cycle.

MEDIUM_4T8

Full FullXR

MC_TapConfiguration_MEDIUM_4T8

Description
The camera requires the Camera Link Medium configuration to deliver 4 8-bit pixels every clock cycle.

MEDIUM_4T10

Full FullXR

MC_TapConfiguration_MEDIUM_4T10

Description
The camera requires the Camera Link Medium configuration to deliver 4 10-bit pixels every clock cycle.

MEDIUM_4T12

Full FullXR

MC_TapConfiguration_MEDIUM_4T12

Description
The camera requires the Camera Link Medium configuration to deliver 4 12-bit pixels every clock cycle.

MEDIUM_6T8

Full FullXR

MC_TapConfiguration_MEDIUM_6T8

Description
The camera requires the Camera Link Medium configuration to deliver 6 8-bit pixels every clock cycle.

FULL_8T8

Full FullXR

MC_TapConfiguration_FULL_8T8

Description
The camera requires the Camera Link Full configuration to deliver 8 8-bit pixels every clock cycle.

DECA_2T40

Full FullXR

MC_TapConfiguration_DECA_2T40

Description
The camera requires the Camera Link 80 Bit (8-tap/10-bit) configuration to deliver 10-bit color components for 2 40-bit pixels every clock cycle.

DECA_3T24

Full FullXR

MC_TapConfiguration_DECA_3T24

Description
The camera requires the Camera Link 72 Bit configuration to deliver 9 8-bit color components for 3 24-bit pixels every clock cycle.

DECA_8T10

Full FullXR

MC_TapConfiguration_DECA_8T10

Description
The camera requires the Camera Link 80 Bit (8-tap/10-bit) configuration to deliver 8 10-bit pixels every clock cycle.

DECA_8T30B3

Full FullXR

MC_TapConfiguration_DECA_8T30B3

Description
The camera requires the Camera Link 80 Bit (8-tap/10-bit) configuration to deliver 24 10-bit color components for 8 30-bit pixels every 3 adjacent clock cycles.

DECA_9T8

Full FullXR

MC_TapConfiguration_DECA_9T8

Description
The camera requires the Camera Link 72 Bit configuration to deliver 9 8-bit pixels every clock cycle.

DECA_10T8

Full FullXR

MC_TapConfiguration_DECA_10T8

Description
The camera requires the Camera Link 80 Bit configuration to deliver 10 8-bit pixels every clock cycle.

LITE_1T8

Base DualBase

MC_TapConfiguration_LITE_1T8

Description
The camera requires the Camera Link Lite configuration to deliver 1 8-bit pixel every clock cycle.

Description

LITE_1T10

Base DualBase

MC_TapConfiguration_LITE_1T10

Description
The camera requires the Camera Link Lite configuration to deliver 1 10-bit pixel every clock cycle.

TapGeometry

Base DualBase Full FullXR

Camera Link tap geometry

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
4273 << 14	TapGeometry	MC_TapGeometry

Parameter Description

This parameter declares the Camera Link tap geometry used by the camera.

Based on this parameter together with TapConfiguration , the frame grabber is able to re-arrange the data in the destination surface.

Refer to "TapGeometry Glossary" on page 486 for terms definitions and naming conventions.

Parameter Values

1X

Base DualBase Full FullXR

MC_TapGeometry_1X

Description
One region along X-axis, 1 tap per region, line-scan camera.

Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W	+1	1	H	+1

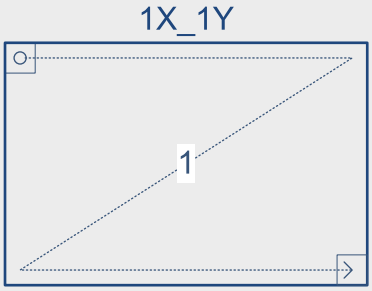
1X_1Y

Base DualBase Full FullXR

MC_TapGeometry_1X_1Y

Description

One region along X-axis, 1 tap per region, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W	+1	1	H	+1

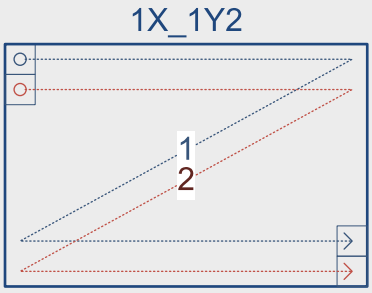
1X_1Y2

Base DualBase Full FullXR

MC_TapGeometry_1X_1Y2

Description

One region along X-axis, 2 vertically adjacent taps per region, line-scan or area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W	+1	1	H-1	+2
Tap#2	1	W	+1	2	H	+2

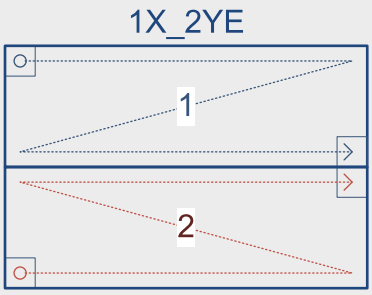
1X_2YE

Base DualBase Full FullXR

MC_TapGeometry_1X_2YE

Description

One region along X-axis, 2 vertical taps per region, start reading from the top/bottom edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W	+1	1	H/2	+1
Tap#2	1	W	+1	H	H/2 + 1	-1

1X2

Base DualBase Full FullXR

MC_TapGeometry_1X2

Description

One region along X-axis, 2 adjacent taps per region, line-scan camera.

Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W-1	+2	1	H	+1
Tap#2	2	W	+2	1	H	+1

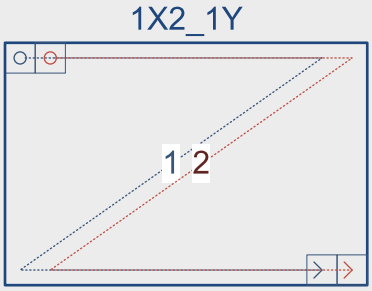
1X2_1Y

Base DualBase Full FullXR

MC_TapGeometry_1X2_1Y

Description

One region along X-axis, 2 adjacent taps per region, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W-1	+2	1	H	+1
Tap#2	2	W	+2	1	H	+1

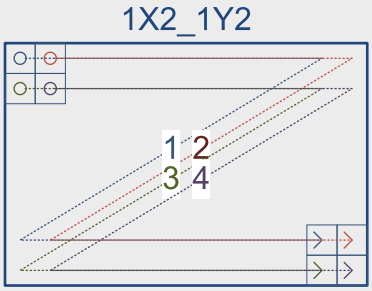
1X2_1Y2

Full FullXR

MC_TapGeometry_1X2_1Y2

Description

One region along X-axis, 2 horizontally adjacent and 2 vertically adjacent taps per region, line-scan or area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W - 1	+2	1	H-1	+2
Tap#2	2	W	+2	1	H-1	+2
Tap#3	1	W - 1	+2	2	H	+2
Tap#4	2	W	+2	2	H	+2

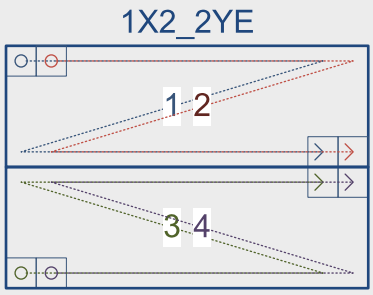
1X2_2YE

Full FullXR

MC_TapGeometry_1X2_2YE

Description

One region along X-axis, 2 horizontally adjacent and 2 vertical taps per region, start reading from the top/bottom edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W - 1	+2	1	H/2	+1
Tap#2	2	W	+2	1	H/2	+1
Tap#3	1	W - 1	+2	H	H/2 + 1	-1
Tap#4	2	W	+2	H	H/2 + 1	-1

1X3

Base DualBase Full FullXR

MC_TapGeometry_1X3

Description

One region along X-axis, 3 adjacent taps per region, line-scan camera.

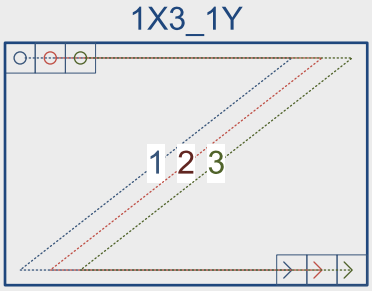
Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W - 2	+3	1	H	+1
Tap#2	2	W - 1	+3	1	H	+1
Tap#3	3	W	+3	1	H	+1

1X3_1Y

Base DualBase Full FullXR

MC_TapGeometry_1X3_1Y

Description
One region along X-axis, 3 adjacent taps per region, area-scan camera.



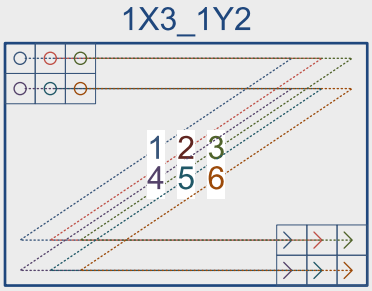
Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W - 2	+3	1	H	+1
Tap#2	2	W - 1	+3	1	H	+1
Tap#3	3	W	+3	1	H	+1

1X3_1Y2

Full FullXR

MC_TapGeometry_1X3_1Y2

Description
One region along X-axis, 3 horizontally adjacent and 2 vertically adjacent taps per region, line-scan or area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W - 2	+3	1	H-1	+2
Tap#2	2	W - 1	+3	1	H-1	+2
Tap#3	3	W	+3	1	H-1	+2
Tap#4	1	W - 2	+3	2	H	+2
Tap#5	2	W - 1	+3	2	H	+2
Tap#6	3	W	+3	2	H	+2

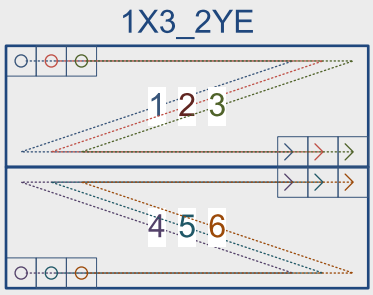
1X3_2YE

Full FullXR

MC_TapGeometry_1X3_2YE

Description

One region along X-axis, 3 horizontally adjacent and 2 vertical taps per region, start reading from the top/bottom edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W - 2	+3	1	H/2	+1
Tap#2	2	W - 1	+3	1	H/2	+1
Tap#3	3	W	+3	1	H/2	+1
Tap#4	1	W - 2	+3	H	H/2 + 1	-1
Tap#5	2	W - 1	+3	H	H/2 + 1	-1
Tap#6	3	W	+3	H	H/2 + 1	-1

1X4

Full FullXR

MC_TapGeometry_1X4

Description

One region along X-axis, 4 adjacent taps per region, line-scan camera.

Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W - 3	+4	1	H	+1
Tap#2	2	W - 2	+4	1	H	+1
Tap#3	3	W - 1	+4	1	H	+1
Tap#4	4	W	+4	1	H	+1

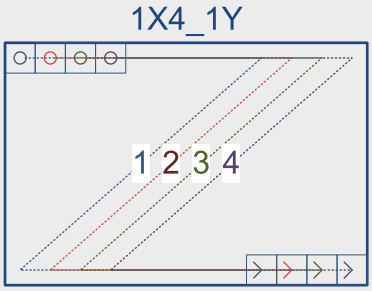
1X4_1Y

Full FullXR

MC_TapGeometry_1X4_1Y

Description

One region along X-axis, 4 adjacent taps per region, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W - 3	+4	1	H	+1
Tap#2	2	W - 2	+4	1	H	+1
Tap#3	3	W - 1	+4	1	H	+1
Tap#4	4	W	+4	1	H	+1

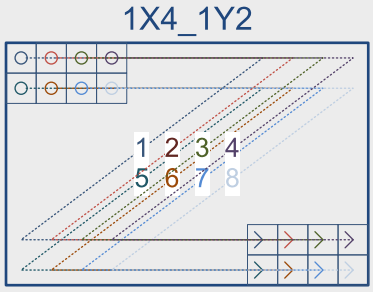
1X4_1Y2

Full FullXR

MC_TapGeometry_1X4_1Y2

Description

One region along X-axis, 4 horizontally adjacent and 2 vertically adjacent taps per region, line-scan or area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W - 3	+4	1	H-1	+2
Tap#2	2	W - 2	+4	1	H-1	+2
Tap#3	3	W - 1	+4	1	H-1	+2
Tap#4	4	W	+4	1	H-1	+2
Tap#5	1	W - 3	+4	2	H	+2
Tap#6	2	W - 2	+4	2	H	+2
Tap#7	3	W - 1	+4	2	H	+2
Tap#8	4	W	+4	2	H	+2

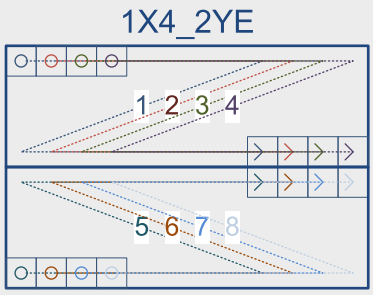
1X4_2YE

Full FullXR

MC_TapGeometry_1X4_2YE

Description

One region along X-axis, 4 horizontally adjacent and 2 vertical taps per region, start reading from the top/bottom edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W - 3	+4	1	H/2	+1
Tap#2	2	W - 2	+4	1	H/2	+1
Tap#3	3	W - 1	+4	1	H/2	+1
Tap#4	4	W	+4	1	H/2	+1
Tap#5	1	W - 3	+4	H	H/2 + 1	-12
Tap#6	2	W - 2	+4	H	H/2 + 1	-12
Tap#7	3	W - 1	+4	H	H/2 + 1	-12
Tap#8	4	W	+4	H	H/2 + 1	-12

1X8

Full FullXR

MC_TapGeometry_1X8

Description

One region along X-axis, 8 adjacent taps per region, line-scan camera.

Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W - 7	+8	1	H	+1
Tap#2	2	W - 6	+8	1	H	+1
Tap#3	3	W - 5	+8	1	H	+1
Tap#4	4	W - 4	+8	1	H	+1
Tap#5	5	W - 3	+8	1	H	+1
Tap#6	6	W - 2	+8	1	H	+1
Tap#7	7	W - 1	+8	1	H	+1
Tap#8	8	W	+8	1	H	+1

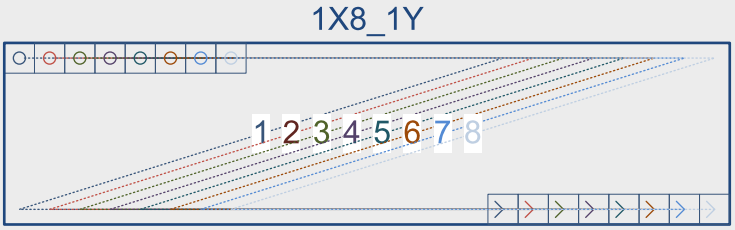
1X8_1Y

Full FullXR

MC_TapGeometry_1X8_1Y

Description

One region along X-axis, 8 adjacent taps per region, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W - 7	+8	1	H	+1
Tap#2	2	W - 6	+8	1	H	+1
Tap#3	3	W - 5	+8	1	H	+1
Tap#4	4	W - 4	+8	1	H	+1
Tap#5	5	W - 3	+8	1	H	+1
Tap#6	6	W - 2	+8	1	H	+1
Tap#7	7	W - 1	+8	1	H	+1
Tap#8	8	W	+8	1	H	+1

1X10

Full FullXR

MC_TapGeometry_1X10

Description

One region along X-axis, 10 adjacent taps per region, line-scan camera.

Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W - 9	+10	1	H	+1
Tap#2	2	W - 8	+10	1	H	+1
Tap#3	3	W - 7	+10	1	H	+1
Tap#4	4	W - 6	+10	1	H	+1
Tap#5	5	W - 5	+10	1	H	+1
Tap#6	6	W - 4	+10	1	H	+1
Tap#7	7	W - 3	+10	1	H	+1
Tap#8	8	W - 2	+10	1	H	+1
Tap#9	9	W - 1	+10	1	H	+1
Tap#10	10	W	+10	1	H	+1

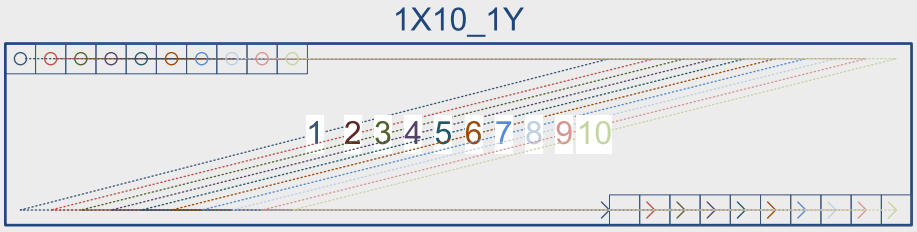
1X10_1Y

Full FullXR

MC_TapGeometry_1X10_1Y

Description

One region along X-axis, 10 adjacent taps per region, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W - 9	+10	1	H	+1
Tap#2	2	W - 8	+10	1	H	+1
Tap#3	3	W - 7	+10	1	H	+1
Tap#4	4	W - 6	+10	1	H	+1
Tap#5	5	W - 5	+10	1	H	+1
Tap#6	6	W - 4	+10	1	H	+1
Tap#7	7	W - 3	+10	1	H	+1
Tap#8	8	W - 2	+10	1	H	+1
Tap#9	9	W - 1	+10	1	H	+1
Tap#10	10	W	+10	1	H	+1

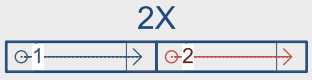
2X

Base DualBase Full FullXR

MC_TapGeometry_2X

Description

Two regions along X-axis, 1 tap per region, line-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2	+1	1	H	+1
Tap#2	W/2 + 1	W	+1	1	H	+1

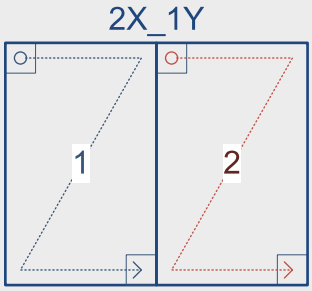
2X_1Y

Base DualBase Full FullXR

MC_TapGeometry_2X_1Y

Description

Two regions along X-axis, 1 tap per region, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2	+1	1	H	+1
Tap#2	W/2 + 1	W	+1	1	H	+1

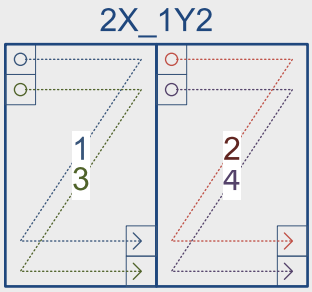
2X_1Y2

Full FullXR

MC_TapGeometry_2X_1Y2

Description

Two regions along X-axis, 2 vertically adjacent taps per region, line-scan or area-scan camera.



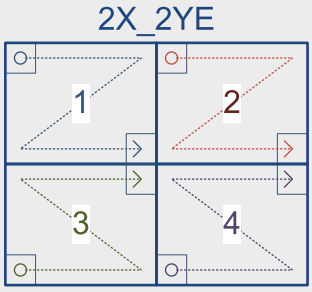
Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2	+1	1	H - 1	+2
Tap#2	W/2 + 1	W	+1	1	H - 1	+2
Tap#3	1	W/2	+1	2	H	+2
Tap#4	W/2 + 1	W	+1	1	H	+2

2X_2YE

Full FullXR

MC_TapGeometry_2X_2YE

Description
Two regions along X-axis, 2 vertical taps per region, start reading from the top/bottom edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2	+1	1	H/2	+1
Tap#2	W/2 + 1	W	+1	1	H/2	+1
Tap#3	1	W/2	+1	H	H/2 + 1	-1
Tap#4	W/2 + 1	W	+1	H	H/2 + 1	-1

2XE

Base DualBase Full FullXR

MC_TapGeometry_2XE

Description
Two regions along X-axis, 1 tap per region, start reading from the left/right edges, line-scan camera.

Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2	+1	1	H	+1
Tap#2	W	W/2 + 1	-1	1	H	+1

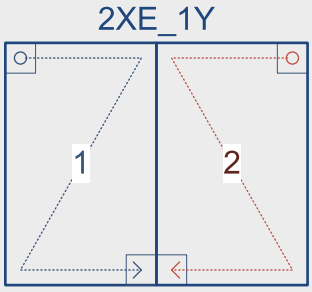
2XE_1Y

Base DualBase Full FullXR

MC_TapGeometry_2XE_1Y

Description

Two regions along X-axis, 1 tap per region, start reading from the left/right edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2	+1	1	H	+1
Tap#2	W	W/2 + 1	-1	1	H	+1

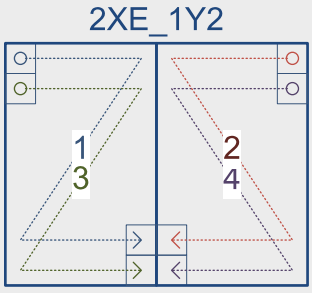
2XE_1Y2

Full FullXR

MC_TapGeometry_2XE_1Y2

Description

Two regions along X-axis, 2 vertically adjacent taps per region, start reading from the left/right edges, line-scan or area-scan camera.



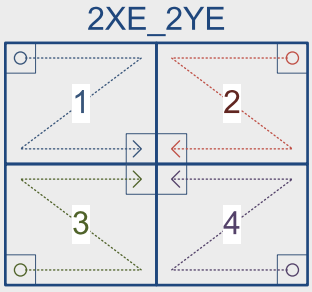
Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2	+1	1	H - 1	+2
Tap#2	W	W/2 + 1	-1	1	H - 1	+2
Tap#3	1	W/2	+1	2	H	+2
Tap#4	W	W/2 + 1	-1	2	H	+2

2XE_2YE

Full FullXR

MC_TapGeometry_2XE_2YE

Description
Two regions along X-axis, 2 vertical taps per region, start reading from the top/bottom edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2	+1	1	H/2	+1
Tap#2	W	W/2 + 1	-1	1	H/2	+1
Tap#3	1	W/2	+1	H	H/2 + 1	-1
Tap#4	W	W/2 + 1	-1	H	H/2 + 1	-1

2XM

Base DualBase Full FullXR

MC_TapGeometry_2XM

Description
Two regions along X-axis, 1 tap per region, start reading from the middle, line-scan camera.

Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/2	1	-1	1	H	+1
Tap#2	W/2 + 1	W	+1	1	H	+1

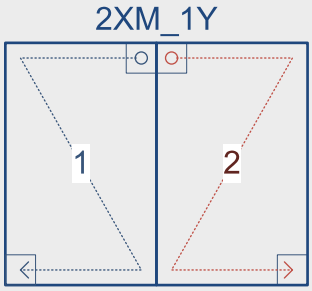
2XM_1Y

Base DualBase Full FullXR

MC_TapGeometry_2XM_1Y

Description

Two regions along X-axis, 1 tap per region, start reading from the middle, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/2	1	-1	1	H	+1
Tap#2	W/2 + 1	W	+1	1	H	+1

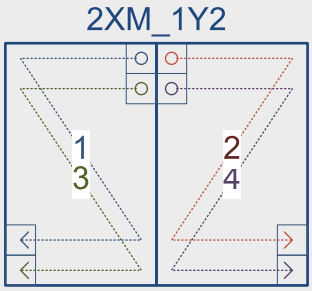
2XM_1Y2

Full FullXR

MC_TapGeometry_2XM_1Y2

Description

Two regions along X-axis, 2 vertically adjacent taps per region, start reading from the middle, line-scan or area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/2	1	-1	1	H - 1	+2
Tap#2	W/2 + 1	W	+1	1	H - 1	+2
Tap#3	W/2	1	-1	2	H	+2
Tap#4	W/2 + 1	W	+1	2	H	+2

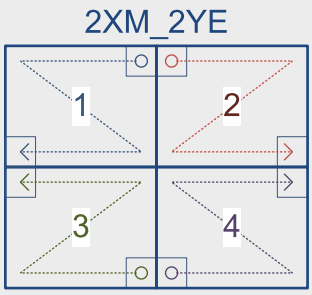
2XM_2YE

Full FullXR

MC_TapGeometry_2XM_2YE

Description

Two regions along X-axis, 2 vertical taps per region, start reading from the middle and from the top/bottom edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/2	1	-1	1	H/2	+1
Tap#2	W/2 + 1	W	+1	1	H/2	+1
Tap#3	W/2	1	-1	H	H/2 + 1	-1
Tap#4	W/2 + 1	W	+1	H	H/2 + 1	-1

2XR

Base DualBase Full FullXR

MC_TapGeometry_2XR

Description

Two regions along X-axis, 1 tap per region, start reading from the right, line-scan camera.

Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/2	1	-1	1	H	+1
Tap#2	W	W/2 + 1	-1	1	H	+1

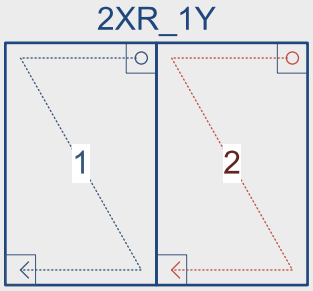
2XR_1Y

Base DualBase Full FullXR

MC_TapGeometry_2XR_1Y

Description

Two regions along X-axis, 1 tap per region, start reading from the right, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/2	1	-1	1	H	+1
Tap#2	W	W/2 + 1	-1	1	H	+1

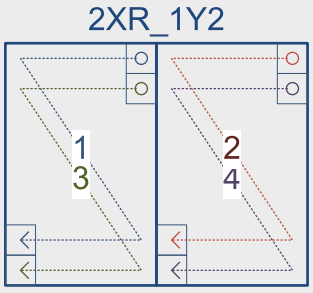
2XR_1Y2

Full FullXR

MC_TapGeometry_2XR_1Y2

Description

Two regions along X-axis, 2 vertically adjacent taps per region, start reading from the right, line-scan or area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/2	1	-1	1	H - 1	+2
Tap#2	W	W/2 + 1	-1	1	H - 1	+2
Tap#3	W/2	1	-1	2	H	+2
Tap#4	W	W/2 + 1	-1	2	H	+2

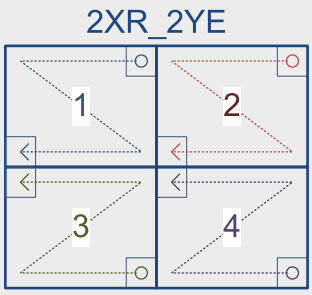
2XR_2YE

Full FullXR

MC_TapGeometry_2XR_2YE

Description

Two regions along X-axis, 2 vertical taps per region, start reading from the right and from the top/bottom edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/2	1	-1	1	H/2	+1
Tap#2	W	W/2 + 1	-1	1	H/2	+1
Tap#3	W/2	1	-1	H	H/2 + 1	-1
Tap#4	W	W/2 + 1	-1	H	H/2 + 1	-1

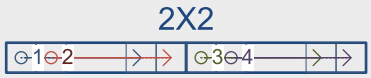
2X2

Full FullXR

MC_TapGeometry_2X2

Description

Two regions along X-axis, 2 adjacent taps per region, line-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2 - 1	+2	1	H	+1
Tap#2	2	W/2	+2	1	H	+1
Tap#3	W/2 + 1	W - 1	+2	1	H	+1
Tap#4	W/2 + 2	W	+2	1	H	+1

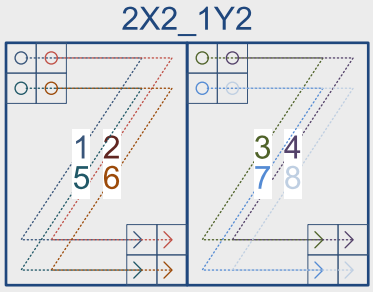
2X2_1Y2

Full FullXR

MC_TapGeometry_2X2_1Y2

Description

Two regions along X-axis, 2 horizontally adjacent and 2 vertically adjacent taps per region, line-scan or area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2 - 1	+2	1	H-1	+2
Tap#2	2	W/2	+2	1	H-1	+2
Tap#3	W/2 + 1	W - 1	+2	1	H-1	+2
Tap#4	W/2 + 2	W	+2	1	H-1	+2
Tap#5	1	W/2 - 1	+2	2	H	+2
Tap#6	2	W/2	+2	2	H	+2
Tap#7	W/2 + 1	W - 1	+2	2	H	+2
Tap#8	W/2 + 2	W	+2	2	H	+2

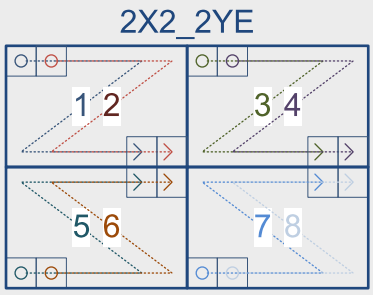
2X2_2YE

Full FullXR

MC_TapGeometry_2X2_2YE

Description

Two regions along X-axis, 2 horizontally adjacent and 2 vertical taps per region, start reading from the top/bottom edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2 - 1	+2	1	H/2	+1
Tap#2	2	W/2	+2	1	H/2	+1
Tap#3	W/2 + 1	W - 1	+2	1	H/2	+1
Tap#4	W/2 + 2	W	+2	1	H/2	+1
Tap#5	1	W/2 - 1	+2	H	H/2 + 1	-1
Tap#6	2	W/2	+2	H	H/2 + 1	-1
Tap#7	W/2 + 1	W - 1	+2	H	H/2 + 1	-1
Tap#8	W/2 + 2	W	+2	H	H/2 + 1	-1

2X2E

Full FullXR

MC_TapGeometry_2X2E

Description

Two regions along X-axis, 2 adjacent taps per region, start reading from the left/right edges, line-scan camera.

Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2 - 1	+2	1	H	+1
Tap#2	2	W/2	+2	1	H	+1
Tap#3	W - 1	W/2 + 1	-2	1	H	+1
Tap#4	W	W/2 + 2	-2	1	H	+1

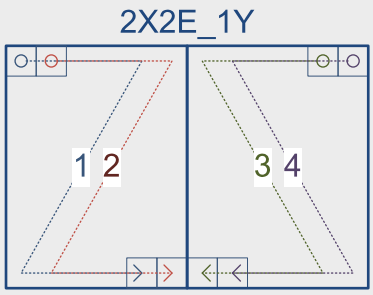
2X2E_1Y

Full FullXR

MC_TapGeometry_2X2E_1Y

Description

Two regions along X-axis, 2 adjacent taps per region, start reading from the left/right edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2 - 1	+2	1	H	+1
Tap#2	2	W/2	+2	1	H	+1
Tap#3	W - 1	W/2 + 1	-2	1	H	+1
Tap#4	W	W/2 + 2	-2	1	H	+1

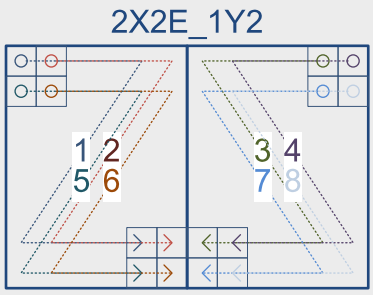
2X2E_1Y2

Full FullXR

MC_TapGeometry_2X2E_1Y2

Description

Two regions along X-axis, 2 horizontally adjacent and 2 vertically adjacent taps per region, start reading from the left/right edges, line-scan or area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2 - 1	+2	1	H-1	+2
Tap#2	2	W/2	+2	1	H-1	+2
Tap#3	W - 1	W/2 + 1	-2	1	H-1	+2
Tap#4	W	W/2 + 2	-2	1	H-1	+2
Tap#5	1	W/2 - 1	+2	2	H	+2
Tap#6	2	W/2	+2	2	H	+2
Tap#7	W - 1	W/2 + 1	-2	2	H	+2
Tap#8	W	W/2 + 2	-2	2	H	+2

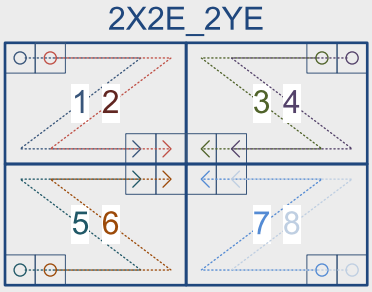
2X2E_2YE

Full FullXR

MC_TapGeometry_2X2E_2YE

Description

Two regions along X-axis, 2 horizontally adjacent and 2 vertical taps per region, start reading from the left/right and top/bottom edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2 - 1	+2	1	H/2	+1
Tap#2	2	W/2	+2	1	H/2	+1
Tap#3	W - 1	W/2 + 1	-2	1	H/2	+1
Tap#4	W	W/2 + 2	-2	1	H/2	+1
Tap#5	1	W/2 - 1	+2	H	H/2 + 1	-1
Tap#6	2	W/2	+2	H	H/2 + 1	-1
Tap#7	W - 1	W/2 + 1	-2	H	H/2 + 1	-1
Tap#8	W	W/2 + 2	-2	H	H/2 + 1	-1

2X2M

Full FullXR

MC_TapGeometry_2X2M

Description

Two regions along X-axis, 2 adjacent taps per region, start reading from the middle, line-scan camera.

Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/2 - 1	1	-2	1	H	+1
Tap#2	W/2	2	-2	1	H	+1
Tap#3	W/2 + 1	W - 1	+2	1	H	+1
Tap#4	W/2 + 2	W	+2	1	H	+1

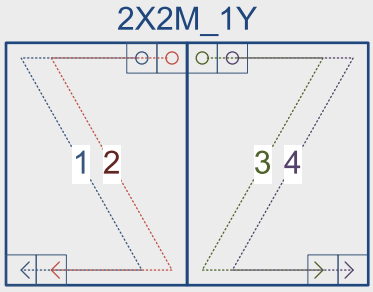
2X2M_1Y

Full FullXR

MC_TapGeometry_2X2M_1Y

Description

Two regions along X-axis, 2 adjacent taps per region, start reading from the middle, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/2 - 1	1	-2	1	H	+1
Tap#2	W/2	2	-2	1	H	+1
Tap#3	W/2 + 1	W - 1	+2	1	H	+1
Tap#4	W/2 + 2	W	+2	1	H	+1

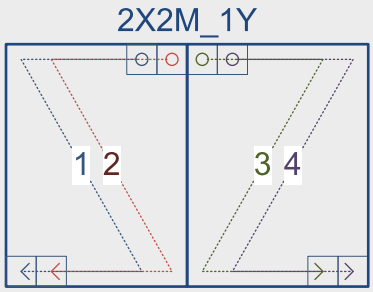
2X2M_1Y

Full FullXR

MC_TapGeometry_2X2M_1Y

Description

Two regions along X-axis, 2 adjacent taps per region, start reading from the middle, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/2 - 1	1	-2	1	H	+1
Tap#2	W/2	2	-2	1	H	+1
Tap#3	W/2 + 1	W - 1	+2	1	H	+1
Tap#4	W/2 + 2	W	+2	1	H	+1

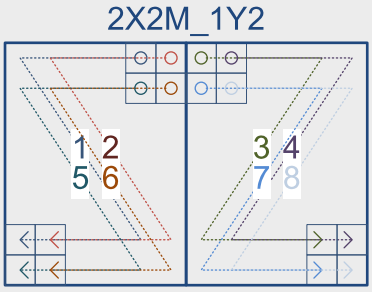
2X2M_1Y2

Full FullXR

MC_TapGeometry_2X2M_1Y2

Description

Two regions along X-axis, 2 horizontally adjacent and 2 vertically adjacent taps per region, start reading from the middle, line-scan or area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/2 - 1	1	-2	1	H-1	+2
Tap#2	W/2	2	-2	1	H-1	+2
Tap#3	W/2 + 1	W - 1	+2	1	H-1	+2
Tap#4	W/2 + 2	W	+2	1	H-1	+2
Tap#5	W/2 - 1	1	-2	2	H	+2
Tap#6	W/2	2	-2	2	H	+2
Tap#7	W/2 + 1	W - 1	+2	2	H	+2
Tap#8	W/2 + 2	W	+2	2	H	+2

2X4

Full FullXR

MC_TapGeometry_2X4

Description

Two regions along X-axis, 4 adjacent taps per region, line-scan camera.

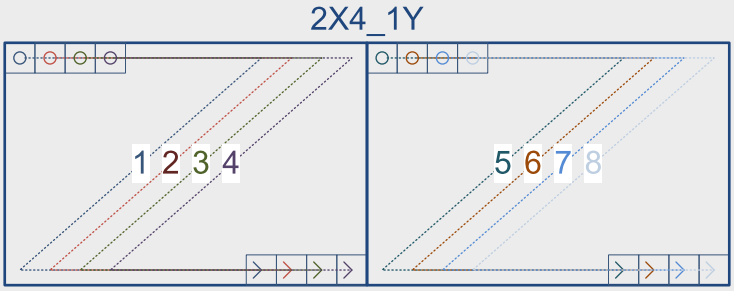
Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2 - 3	+4	1	H	+1
Tap#2	2	W/2 - 2	+4	1	H	+1
Tap#3	3	W/2 - 1	+4	1	H	+1
Tap#4	4	W/2	+4	1	H	+1
Tap#5	W/2 + 1	W - 3	+4	1	H	+1
Tap#6	W/2 + 2	W - 2	+4	1	H	+1
Tap#7	W/2 + 3	W - 1	+4	1	H	+1
Tap#8	W/2 + 4	W	+4	1	H	+1

2X4_1Y

Full FullXR

MC_TapGeometry_2X4_1Y

Description
Two regions along X-axis, 4 adjacent taps per region, area-scan camera.



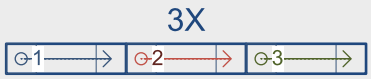
Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/2 - 3	+4	1	H	+1
Tap#2	2	W/2 - 2	+4	1	H	+1
Tap#3	3	W/2 - 1	+4	1	H	+1
Tap#4	4	W/2	+4	1	H	+1
Tap#5	W/2 + 1	W - 3	+4	1	H	+1
Tap#6	W/2 + 2	W - 2	+4	1	H	+1
Tap#7	W/2 + 3	W - 1	+4	1	H	+1
Tap#8	W/2 + 4	W	+4	1	H	+1

3X

Base DualBase Full FullXR

MC_TapGeometry_3X

Description
Three regions along X-axis, 1 tap per region, line-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/3	+1	1	H	+1
Tap#2	W/3 + 1	2W/3	+1	1	H	+1
Tap#3	2W/3 + 1	W	+1	1	H	+1

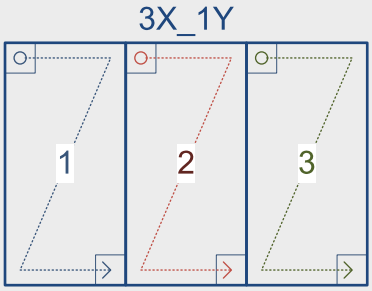
3X_1Y

Base DualBase Full FullXR

MC_TapGeometry_3X_1Y

Description

Three regions along X-axis, 1 tap per region, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/3	+1	1	H	+1
Tap#2	W/3 + 1	2W/3	+1	1	H	+1
Tap#3	2W/3 + 1	W	+1	1	H	+1

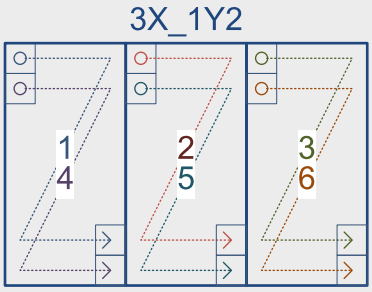
3X_1Y2

Full FullXR

MC_TapGeometry_3X_1Y2

Description

Three regions along X-axis, 2 vertically adjacent taps per region, line-scan or area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/3	+1	1	H-1	+2
Tap#2	W/3 + 1	2W/3	+1	1	H-1	+2
Tap#3	2W/3 + 1	W	+1	1	H-1	+2
Tap#4	1	W/3	+1	2	H	+2
Tap#5	W/3 + 1	2W/3	+1	2	H	+2
Tap#6	2W/3 + 1	W	+1	2	H	+2

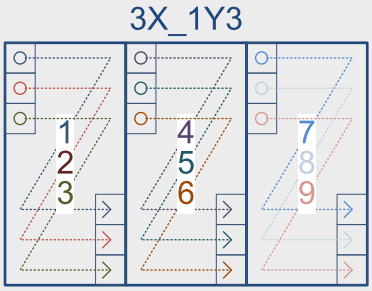
3X_1Y3

Full FullXR

MC_TapGeometry_3X_1Y3

Description

Three regions along X-axis, 3 vertically adjacent taps per region, line-scan or area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/3	+1	1	H-2	+3
Tap#2	1	W/3	+1	2	H-1	+3
Tap#3	1	W/3	+1	3	H	+3
Tap#4	W/3 + 1	2W/3	+1	1	H-2	+3
Tap#5	W/3 + 1	2W/3	+1	2	H-1	+3
Tap#6	W/3 + 1	2W/3	+1	3	H	+3
Tap#7	2W/3 + 1	W	+1	1	H-2	+3
Tap#8	2W/3 + 1	W	+1	2	H-1	+3
Tap#9	2W/3 + 1	W	+1	3	H	+3

4X

Full FullXR

MC_TapGeometry_4X

Description

Four regions along X-axis, 1 tap per region, line-scan camera.

Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/4	+1	1	H	+1
Tap#2	W/4 + 1	W/2	+1	1	H	+1
Tap#3	W/2 + 1	3W/4	+1	1	H	+1
Tap#4	3W/4 + 1	W	+1	1	H	+1

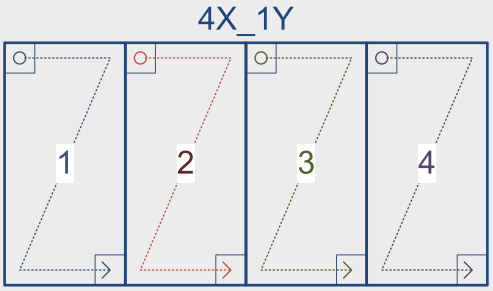
4X_1Y

Full FullXR

MC_TapGeometry_4X_1Y

Description

Four regions along X-axis, 1 tap per region, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/4	+1	1	H	+1
Tap#2	W/4 + 1	W/2	+1	1	H	+1
Tap#3	W/2 + 1	3W/4	+1	1	H	+1
Tap#4	3W/4 + 1	W	+1	1	H	+1

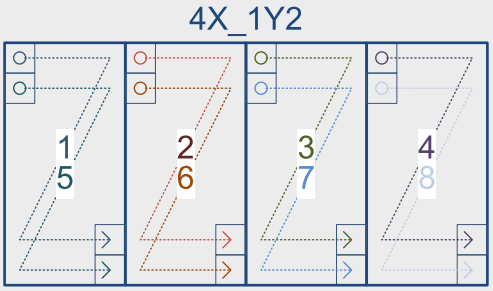
4X_1Y2

Full FullXR

MC_TapGeometry_4X_1Y2

Description

Four regions along X-axis, 2 vertically adjacent taps per region, line-scan or area-scan camera.



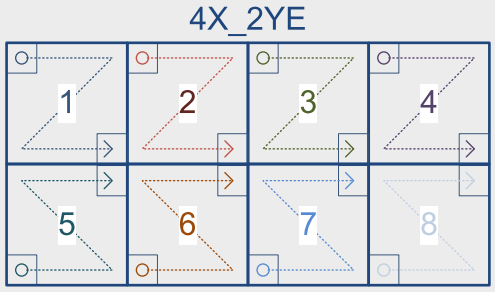
Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/4	+1	1	H - 1	+2
Tap#2	W/4 + 1	W/2	+1	1	H - 1	+2
Tap#3	W/2 + 1	3W/4	+1	1	H - 1	+2
Tap#4	3W/4 + 1	W	+1	1	H - 1	+2
Tap#5	1	W/4	+1	2	H	+2
Tap#6	W/4 + 1	W/2	+1	2	H	+2
Tap#7	W/2 + 1	3W/4	+1	2	H	+2
Tap#8	3W/4 + 1	W	+1	2	H	+2

4X_2YE

Full FullXR

MC_TapGeometry_4X_2YE

Description
Four regions along X-axis, 2 vertical taps per region, start reading from the top/bottom edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/4	+1	1	H/2	+1
Tap#2	W/4 + 1	W/2	+1	1	H/2	+1
Tap#3	W/2 + 1	3W/4	+1	1	H/2	+1
Tap#4	3W/4 + 1	W	+1	1	H/2	+1
Tap#5	1	W/4	+1	H	H/2 + 1	-1
Tap#6	W/4 + 1	W/2	+1	H	H/2 + 1	-1
Tap#7	W/2 + 1	3W/4	+1	H	H/2 + 1	-1
Tap#8	3W/4 + 1	W	+1	H	H/2 + 1	-1

4XE

Full FullXR

MC_TapGeometry_4XE

Description
Four regions along X-axis, 1 tap per region, start reading from the left/right edges, line-scan camera.

Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/4	+1	1	H	+1
Tap#2	W/4 + 1	W/2	+1	1	H	+1
Tap#3	3W/4	W/2 + 1	-1	1	H	+1
Tap#4	W	3W/4 + 1	-1	1	H	+1

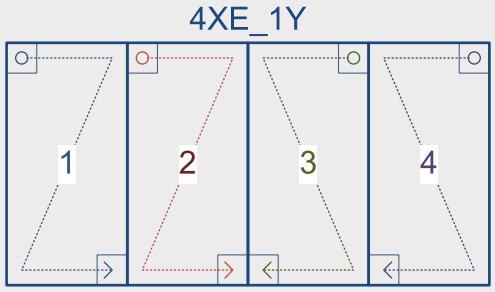
4XE_1Y

Full FullXR

MC_TapGeometry_4XE_1Y

Description

Four regions along X-axis, 1 tap per region, start reading from the left/right edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/4	+1	1	H	+1
Tap#2	W/4 + 1	W/2	+1	1	H	+1
Tap#3	3W/4	W/2 + 1	-1	1	H	+1
Tap#4	W	3W/4 + 1	-1	1	H	+1

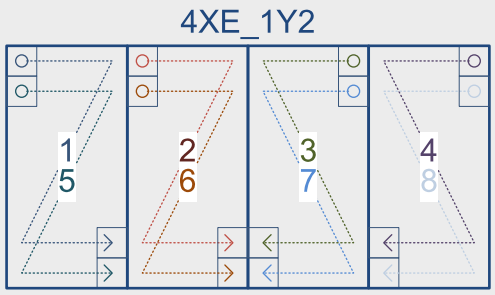
4XE_1Y2

Full FullXR

MC_TapGeometry_4XE_1Y2

Description

Four regions along X-axis, 2 vertically adjacent taps per region, start reading from the left/right edges, line-scan or area-scan camera.



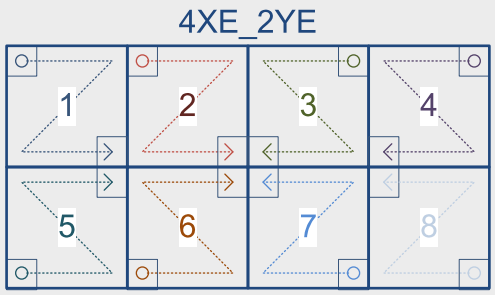
Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/4	+1	1	H - 1	+2
Tap#2	W/4 + 1	W/2	+1	1	H - 1	+2
Tap#3	3W/4	W/2 + 1	-1	1	H - 1	+2
Tap#4	W	3W/4 + 1	-1	1	H - 1	+2
Tap#5	1	W/4	+1	2	H	+2
Tap#6	W/4 + 1	W/2	+1	2	H	+2
Tap#7	3W/4	W/2 + 1	-1	2	H	+2
Tap#8	W	3W/4 + 1	-1	2	H	+2

4XE_2YE

Full FullXR

MC_TapGeometry_4XE_2YE

Description
Four regions along X-axis, 2 vertical taps per region, start reading from the top/bottom edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/4	+1	1	H/2	+1
Tap#2	W/4 + 1	W/2	+1	1	H/2	+1
Tap#3	3W/4	W/2 + 1	-1	1	H/2	+1
Tap#4	W	3W/4 + 1	-1	1	H/2	+1
Tap#5	1	W/4	+1	H	H/2 + 1	-1
Tap#6	W/4 + 1	W/2	+1	H	H/2 + 1	-1
Tap#7	3W/4	W/2 + 1	-1	H	H/2 + 1	-1
Tap#8	W	3W/4 + 1	-1	H	H/2 + 1	-1

4XR

Full FullXR

MC_TapGeometry_4XR

Description
Four regions along X-axis, 1 tap per region, start reading from the right, line-scan camera.

Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/4	1	-1	1	H	+1
Tap#2	W/2	W/4 + 1	-1	1	H	+1
Tap#3	3W/4	W/2 + 1	-1	1	H	+1
Tap#4	W	3W/4 + 1	-1	1	H	+1

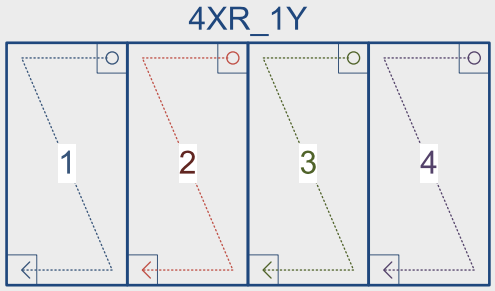
4XR_1Y

Full FullXR

MC_TapGeometry_4XR_1Y

Description

Four regions along X-axis, 1 tap per region, start reading from the right, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/4	1	-1	1	H	+1
Tap#2	W/2	W/4 + 1	-1	1	H	+1
Tap#3	3W/4	W/2 + 1	-1	1	H	+1
Tap#4	W	3W/4 + 1	-1	1	H	+1

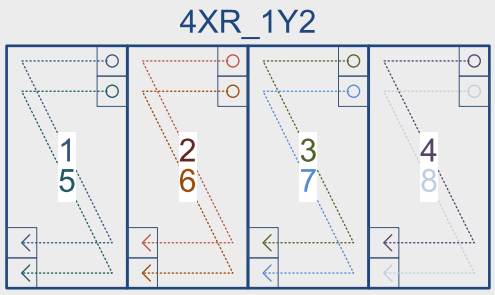
4XR_1Y2

Full FullXR

MC_TapGeometry_4XR_1Y2

Description

Four regions along X-axis, 2 vertically adjacent taps per region, start reading from the right, line-scan or area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/4	1	-1	1	H - 1	+2
Tap#2	W/2	W/4 + 1	-1	1	H - 1	+2
Tap#3	3W/4	W/2 + 1	-1	1	H - 1	+2
Tap#4	W	3W/4 + 1	-1	1	H - 1	+2
Tap#5	W/4	1	-1	2	H	+2
Tap#6	W/2	W/4 + 1	-1	2	H	+2
Tap#7	3W/4	W/2 + 1	-1	2	H	+2
Tap#8	W	3W/4 + 1	-1	2	H	+2

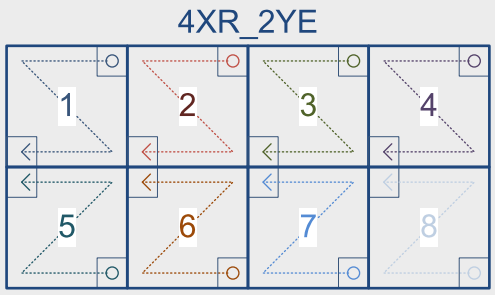
4XR_2YE

Full FullXR

MC_TapGeometry_4XR_2YE

Description

Four regions along X-axis, 2 vertical taps per region, start reading from the right and from the top/bottom edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/4	1	-1	1	H/2	+1
Tap#2	W/2	W/4 + 1	-1	1	H/2	+1
Tap#3	3W/4	W/2 + 1	-1	1	H/2	+1
Tap#4	W	3W/4 + 1	-1	1	H/2	+1
Tap#5	W/4	1	-1	H	H/2 + 1	-1
Tap#6	W/2	W/4 + 1	-1	H	H/2 + 1	-1
Tap#7	3W/4	W/2 + 1	-1	H	H/2 + 1	-1
Tap#8	W	3W/4 + 1	-1	H	H/2 + 1	-1

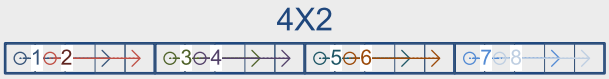
4X2

Full FullXR

MC_TapGeometry_4X2

Description

Four regions along X-axis, 2 adjacent taps per region, line-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/4 - 1	+2	1	H	+1
Tap#2	2	W/4	+2	1	H	+1
Tap#3	W/4 + 1	W/2 - 1	+2	1	H	+1
Tap#4	W/4 + 2	W/2	+2	1	H	+1
Tap#5	W/2 + 1	3W/4 - 1	+2	1	H	+1
Tap#6	W/2 + 2	3W/4	+2	1	H	+1
Tap#7	3W/4 + 1	W - 1	+2	1	H	+1
Tap#8	3W/4 + 2	W	+2	1	H	+1

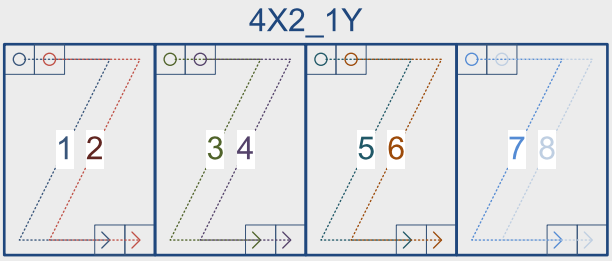
4X2_1Y

Full FullXR

MC_TapGeometry_4X2_1Y

Description

Four regions along X-axis, 2 adjacent taps per region, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/4 - 1	+2	1	H	+1
Tap#2	2	W/4	+2	1	H	+1
Tap#3	W/4 + 1	W/2 - 1	+2	1	H	+1
Tap#4	W/4 + 2	W/2	+2	1	H	+1
Tap#5	W/2 + 1	3W/4 - 1	+2	1	H	+1
Tap#6	W/2 + 2	3W/4	+2	1	H	+1
Tap#7	3W/4 + 1	W - 1	+2	1	H	+1
Tap#8	3W/4 + 2	W	+2	1	H	+1

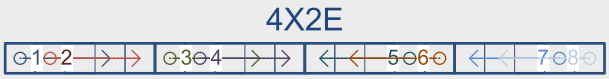
4X2E

Full FullXR

MC_TapGeometry_4X2E

Description

Four regions along X-axis, 2 adjacent taps per region, start reading from the left/right edges, line-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/4 - 1	+2	1	H	+1
Tap#2	2	W/4	+2	1	H	+1
Tap#3	W/4 + 1	W/2 - 1	+2	1	H	+1
Tap#4	W/4 + 2	W/2	+2	1	H	+1
Tap#5	3W/4 - 1	W/2 + 1	-2	1	H	+1
Tap#6	3W/4	W/2 + 2	-2	1	H	+1
Tap#7	W - 1	3W/4 + 1	-2	1	H	+1
Tap#8	W	3W/4 + 2	-2	1	H	+1

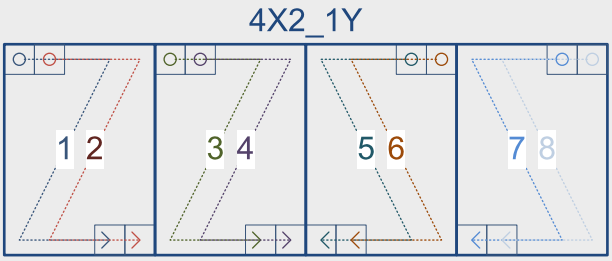
4X2E_1Y

Full FullXR

MC_TapGeometry_4X2E_1Y

Description

Four regions along X-axis, 2 adjacent taps per region, start reading from the left/right edges, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/4 - 1	+2	1	H	+1
Tap#2	2	W/4	+2	1	H	+1
Tap#3	W/4 + 1	W/2 - 1	+2	1	H	+1
Tap#4	W/4 + 2	W/2	+2	1	H	+1
Tap#5	3W/4 - 1	W/2 + 1	-2	1	H	+1
Tap#6	3W/4	W/2 + 2	-2	1	H	+1
Tap#7	W - 1	3W/4 + 1	-2	1	H	+1
Tap#8	W	3W/4 + 2	-2	1	H	+1

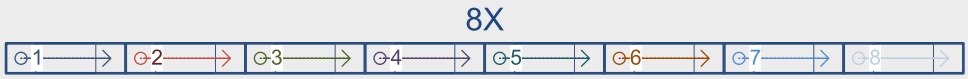
8X

Full FullXR

MC_TapGeometry_8X

Description

Eight regions along X-axis, 1 tap per region, line-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/8	+1	1	H	+1
Tap#2	W/8 + 1	2W/8	+1	1	H	+1
Tap#3	2W/8 + 1	3W/8	+1	1	H	+1
Tap#4	3W/8 + 1	4W/8	+1	1	H	+1
Tap#5	4W/8 + 1	5W/8	+1	1	H	+1
Tap#6	5W/8 + 1	6W/8	+1	1	H	+1
Tap#7	6W/8 + 1	7W/8	+1	1	H	+1
Tap#8	7W/8 + 1	W	+1	1	H	+1

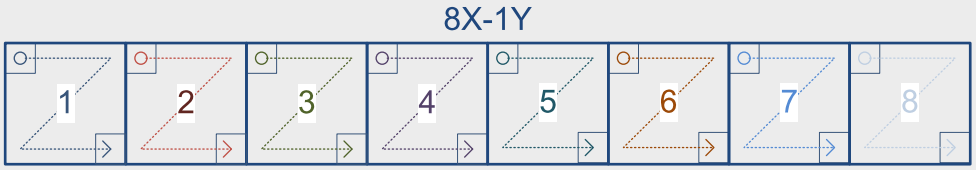
8X_1Y

Full FullXR

MC_TapGeometry_8X_1Y

Description

Eight regions along X-axis, 1 tap per region, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/8	+1	1	H	+1
Tap#2	W/8 + 1	2W/8	+1	1	H	+1
Tap#3	2W/8 + 1	3W/8	+1	1	H	+1
Tap#4	3W/8 + 1	4W/8	+1	1	H	+1
Tap#5	4W/8 + 1	5W/8	+1	1	H	+1
Tap#6	5W/8 + 1	6W/8	+1	1	H	+1
Tap#7	6W/8 + 1	7W/8	+1	1	H	+1
Tap#8	7W/8 + 1	W	+1	1	H	+1

8XR

Full FullXR

MC_TapGeometry_8XR

Description

Eight regions along X-axis, 1 tap per region, start reading from the right, line-scan camera.

Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/8	1	-1	1	H	+1
Tap#2	2W/8	W/8 + 1	-1	1	H	+1
Tap#3	3W/8	2W/8 + 1	-1	1	H	+1
Tap#4	4W/8	3W/8 + 1	-1	1	H	+1
Tap#5	5W/8	4W/8 + 1	-1	1	H	+1
Tap#6	6W/8	5W/8 + 1	-1	1	H	+1
Tap#7	7W/8	6W/8 + 1	-1	1	H	+1
Tap#8	W	7W/8 + 1	-1	1	H	+1

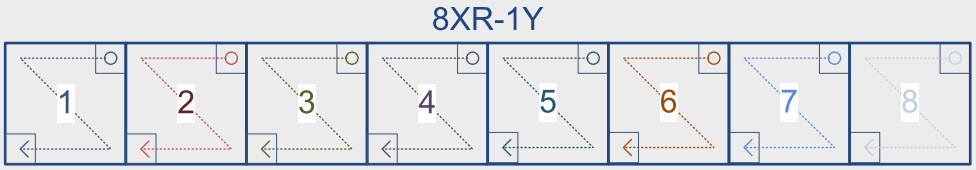
8XR_1Y

Full FullXR

MC_TapGeometry_8XR_1Y

Description

Eight regions along X-axis, 1 tap per region, start reading from the right, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	W/8	1	-1	1	H	+1
Tap#2	2W/8	W/8 + 1	-1	1	H	+1
Tap#3	3W/8	2W/8 + 1	-1	1	H	+1
Tap#4	4W/8	3W/8 + 1	-1	1	H	+1
Tap#5	5W/8	4W/8 + 1	-1	1	H	+1
Tap#6	6W/8	5W/8 + 1	-1	1	H	+1
Tap#7	7W/8	6W/8 + 1	-1	1	H	+1
Tap#8	W	7W/8 + 1	-1	1	H	+1

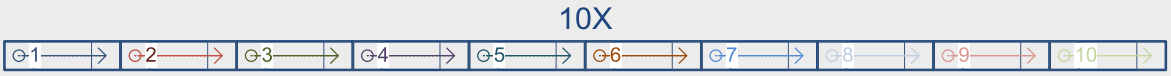
10X

Full FullXR

MC_TapGeometry_10X

Description

Ten regions along X-axis, 1 tap per region, line-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/10	+1	1	H	+1
Tap#2	W/10 + 1	2W/10	+1	1	H	+1
Tap#3	2W/10 + 1	3W/10	+1	1	H	+1
Tap#4	3W/10 + 1	4W/10	+1	1	H	+1
Tap#5	4W/10 + 1	5W/10	+1	1	H	+1
Tap#6	5W/10 + 1	6W/10	+1	1	H	+1
Tap#7	6W/10 + 1	7W/10	+1	1	H	+1
Tap#8	7W/10 + 1	8W/10	+1	1	H	+1
Tap#9	8W/10 + 1	9W/10	+1	1	H	+1
Tap#10	9W/10 + 1	W	+1	1	H	+1

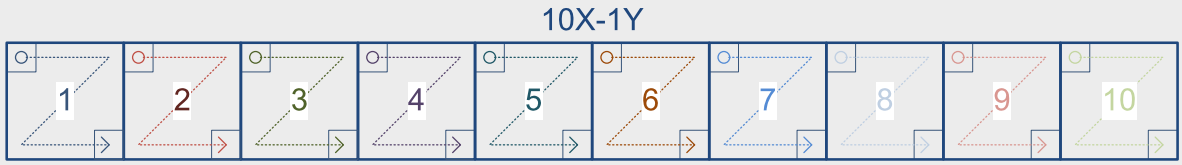
10X_1Y

Full FullXR

MC_TapGeometry_10X_1Y

Description

Ten regions along X-axis, 1 tap per region, area-scan camera.



Tap#	X Start	X End	Step X	Y Start	Y End	Step Y
Tap#1	1	W/10	+1	1	H	+1
Tap#2	W/10 + 1	2W/10	+1	1	H	+1
Tap#3	2W/10 + 1	3W/10	+1	1	H	+1
Tap#4	3W/10 + 1	4W/10	+1	1	H	+1
Tap#5	4W/10 + 1	5W/10	+1	1	H	+1
Tap#6	5W/10 + 1	6W/10	+1	1	H	+1
Tap#7	6W/10 + 1	7W/10	+1	1	H	+1
Tap#8	7W/10 + 1	8W/10	+1	1	H	+1
Tap#9	8W/10 + 1	9W/10	+1	1	H	+1
Tap#10	9W/10 + 1	W	+1	1	H	+1

ColorMethod

Base DualBase Full FullXR

Method used at sensor level to build color information

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1010 << 14	ColorMethod	MC_ColorMethod

Parameter Description

This parameter returns the way the color information is built. See also "Camera Color Analysis Method" on page 503.

Parameter Values

NONE

Base DualBase Full FullXR

MC_ColorMethod_NONE

Description
The camera is monochrome.

RGB

Base DualBase Full FullXR

MC_ColorMethod_RGB

Description
The camera uses a coated sensor and an internal processor to reconstruct the full color information. The color information is available as three R, G, B video data streams.

PRISM

Base DualBase Full FullXR

MC_ColorMethod_PRISM

Description
The camera uses a wavelength-separating prism to feed three distinct imaging sensors. The color information is available as three R, G, B video data streams.

TRILINEAR

Base DualBase Full FullXR

MC_ColorMethod_TRILINEAR

Description
The camera uses three parallel sensing linear arrays of pixels exhibiting different wavelength sensitivities. The color information is available as three R, G, B video data streams.

BAYER

Base DualBase Full FullXR

MC_ColorMethod_BAYER

Description
The camera uses a single imaging sensor coated with a special wavelength-separating patterned filter. The color information is available as a single video data stream embedding the RGB information.

ColorRegistration

Base DualBase Full FullXR

Alignment of the color pattern filter over the camera window

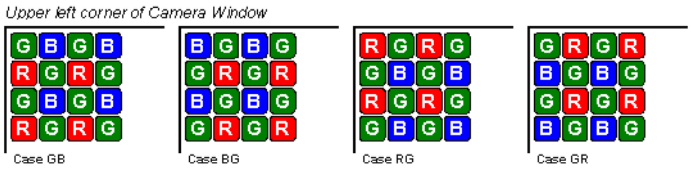
Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1273 << 14	ColorRegistration	MC_ColorRegistration

Parameter Description

When ColorMethod is BAYER, this parameter indicates how the Bayer pattern filter covers the camera active window.



Upper left corner of camera window

When ColorMethod is TRILINEAR, this parameter states the order the three sensing lines are arranged on the CCD chip.

This parameter is otherwise irrelevant.

See also "Camera Color Pattern Filter Alignment" on page 503.

Parameter Values

GB

Base DualBase Full FullXR

MC_ColorRegistration_GB
Description The first two pixels are green and blue.
Applicability condition(s) *Condition:* ColorMethod is set to BAYER

BG

Base DualBase Full FullXR

MC_ColorRegistration_BG

Description
The first two pixels are blue and green.

Applicability condition(s)
Condition: ColorMethod is set to BAYER

RG

Base DualBase Full FullXR

MC_ColorRegistration_RG

Description
The first two pixels are red and green.

Applicability condition(s)
Condition: ColorMethod is set to BAYER

GR

Base DualBase Full FullXR

MC_ColorRegistration_GR

Description
The first two pixels are green and red.

Applicability condition(s)
Condition: ColorMethod is set to BAYER

RGB

Base DualBase Full FullXR

MC_ColorRegistration_RGB

Description
The three sensing lines are ordered as red, green and blue.

Applicability condition(s)
Condition: ColorMethod is set to TRILINEAR

GBR

Base DualBase Full FullXR

MC_ColorRegistration_GBR

Description
The three sensing lines are ordered as green, blue and red.

Applicability condition(s)
Condition: ColorMethod is set to TRILINEAR

BRG

Base DualBase Full FullXR

MC_ColorRegistration_BRG

Description
The three sensing lines are ordered as blue, red and green.

Applicability condition(s)
Condition: ColorMethod is set to TRILINEAR

ColorRegistrationControl

Base DualBase Full FullXR

Controls the color registration

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10850 << 14	ColorRegistrationControl	MC_ColorRegistrationControl

Parameter Description

This enumerated parameter controls the method used for ensuring the correct registration of the colors in the image.

Parameter Usage

For the particular case of Bayer CFA bilinear line-scan cameras such as the Basler Sprint color, it is necessary to start the acquisition of a new image at a 2-line boundary to ensure a correct color registration of the captured Bayer CFA image.

This is achieved by using the FVAL signal to discriminate between the first and the second line of the Bayer CFA sensor.

Parameter Values

FVAL

Base DualBase Full FullXR

MC_ColorRegistrationControl_FVAL
Description Use the FVAL signal as a qualifier for the first line of an image. The first line of an image always corresponds to the first LVAL after FVAL rising.
Applicability condition(s) *Condition:* All BoardTopology values but MONO_SLOW and DUO_SLOW

NONE

Base DualBase Full FullXR

MC_ColorRegistrationControl_NONE
Description Ignore any signal for qualifying the first line of an image (in line-scan acquisition).
Default value.

ExposeOverlap

Base DualBase Full FullXR

Status of the expose to read-out relationship

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1001 << 14	ExposeOverlap	MC_ExposeOverlap

Parameter Description

This parameter indicates whether the expose condition is allowed to overlap the previous read-out condition. This applies to line-scan and area-scan cameras.

ExposeOverlap is always allowed for line-scan cameras.

Parameter Values

ALLOW

Base DualBase Full FullXR

MC_ExposeOverlap_ALLOW
Description The expose condition is allowed to overlap the previous read-out condition.

Expose

Base DualBase Full FullXR

Camera exposure principle

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1011 << 14	Expose	MC_Expose

Parameter Description

This parameter declares the exposure principle of the camera. The camera exposure principle is the way the light exposure function is handled inside the camera. This equally applies to area-scan and line-scan camera.

Parameter Values

PLSTRG

Base DualBase Full FullXR

MC_Expose_PLSTRG

Description
The line or frame exposure condition starts upon receiving a pulse from the frame grabber.

WIDTH

Base DualBase Full FullXR

MC_Expose_WIDTH

Description
The duration of a pulse issued by the frame grabber determines the line or frame exposure condition.

INTCTL

Base DualBase Full FullXR

MC_Expose_INTCTL

Description
The line or frame exposure condition is totally controlled by the camera. The exposure duration is set through camera configuration settings.

INTPRM

Base DualBase Full FullXR

MC_Expose_INTPRM

Description
The exposure is permanent.

Readout

Base DualBase Full FullXR

Camera read-out principle

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1012 << 14	Readout	MC_Readout

Parameter Description

This parameter declares the read-out principle of the camera. The camera read-out principle is the way the read-out function is handled inside the camera.

Parameter Values

PLSTRG

Base DualBase Full FullXR

MC_Readout_PLSTRG

Base DualBase Full FullXR

Description

With line-scan cameras, the line read-out condition starts upon receiving a pulse from the frame grabber.

INTCTL

Base DualBase Full FullXR

MC_Readout_INTCTL

Base DualBase Full FullXR

Description

With line-scan cameras, the read-out duration is set through camera configuration settings. With area-scan cameras, the line read-out condition is totally controlled by the camera.

ResetCtl

Base DualBase Full FullXR

Electrical style of main reset control line to camera

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1027 << 14	ResetCtl	MC_ResetCtl

Parameter Description

This parameter, along with ResetEdge, declares the attributes of the main reset signal applied to the camera feeding the channel.

In case of area-scan cameras, the main reset signal implements the asynchronous frame reset function, which usually triggers the frame exposure condition inside the camera.

In case of line-scan cameras, the main reset signal implements the line reset function, which usually triggers the line exposure condition inside the camera.

Some cameras use an additional reset control line to independently control the expose and read-out functions. Refer to AuxResetCtl .

Parameter Values

NONE

Base DualBase Full FullXR

MC_ResetCtl_NONE
Description The camera has no reset control line.

DIFF

Base DualBase Full FullXR

MC_ResetCtl_DIFF
Description The camera reset control line requires a signal at RS-422 or LVDS differential levels.

ResetEdge

Base DualBase Full FullXR

Significant edge of main reset control line to camera

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1028 << 14	ResetEdge	MC_ResetEdge

Parameter Description

This parameter, along with ResetCtl , declares the attributes of the main reset signal applied to the camera feeding the channel.

In case of area-scan cameras, the main reset signal implements the asynchronous frame reset function, which usually triggers the frame exposure condition inside the camera.

In case of line-scan cameras, the main reset signal implements the line reset function, which usually triggers the line exposure condition inside the camera.

Some cameras use an additional reset control line to independently control the expose and read-out functions. Refer to the AuxResetEdge parameter.

The parameter indicates the logic polarity delivered through the main reset line the camera obeys to.

Parameter Values

GOHIGH

Base DualBase Full FullXR

MC_ResetEdge_GOHIGH
Description The camera reacts to a positive going pulse over the main reset control line.

GOLOW

Base DualBase Full FullXR

MC_ResetEdge_GOLOW
Description The camera reacts to a negative going pulse over the main reset control line.

AuxResetCtl

Base DualBase Full FullXR

Electrical style of auxiliary reset control line to camera

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1029 << 14	AuxResetCtl	MC_AuxResetCtl

Parameter Description

This parameter, along with AuxResetEdge , declares the attributes of the auxiliary reset signal applied to the camera feeding the channel.

Some cameras (area-scan or line-scan) use two reset control lines to independently control the expose and read-out functions. Refer to the ResetCtl parameter.

Parameter Values

NONE

Base DualBase Full FullXR

MC_AuxResetCtl_NONE
Description The camera has no auxiliary reset control line.

DIFF

Base DualBase Full FullXR

MC_AuxResetCtl_DIFF
Description The camera auxiliary reset control line requires a signal at RS-422 or LVDS differential levels.

AuxResetEdge

Base DualBase Full FullXR

Significant edge of auxiliary reset control line to camera

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1030 << 14	AuxResetEdge	MC_AuxResetEdge

Parameter Description

This parameter, along with AuxResetCtl , declares the attributes of the auxiliary reset signal applied to the camera feeding the channel.

Some cameras (area-scan or line-scan) use two reset control lines to independently control the expose and read-out functions. Refer to the ResetCtl parameter.

The parameter indicates the logic polarity delivered through the auxiliary reset line the camera obeys to.

Parameter Values

GOHIGH

Base DualBase Full FullXR

MC_AuxResetEdge_GOHIGH
Description The camera reacts to a positive going pulse over the auxiliary reset control line.

GOLOW

Base DualBase Full FullXR

MC_AuxResetEdge_GOLOW
Description The camera reacts to a negative going pulse over the auxiliary reset control line.

ResetDur

Base DualBase Full FullXR

Required duration of pulse sent through reset control line to camera

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
1031 << 14	ResetDur	MC_ResetDur

Parameter Description

This parameter declares the minimum pulse width to be applied to the reset control line for the camera to assume the proper reaction.

It characterizes both the main (RESET) and auxiliary reset (AUXRESET) control lines when it exists.

With area-scan cameras, ResetDur is expressed as a number of video lines.

With line-scan cameras, ResetDur is expressed in nanoseconds.

ExposeMin_us

Base DualBase Full FullXR

Minimum duration of grabber-controlled exposure allowed by camera, expressed in microseconds

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
1033 << 14	ExposeMin_us	MC_ExposeMin_us

Parameter Description

This parameter applies to area-scan camera operated in CTL mode, stating the minimum tolerated duration of the frame exposure duration as specified by the camera manufacturer.

It also applies to most the line-scan cameras, stating the minimum tolerated duration of the line exposure duration as specified by the camera manufacturer.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 microsecond *Minimum range value.*
5000000	5,000,000 microseconds (=5 seconds) *Maximum range value.*

ExposeMax_us

Base DualBase Full FullXR

Maximum duration of grabber-controlled exposure allowed by camera, expressed in microseconds

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
1034 << 14	ExposeMax_us	MC_ExposeMax_us

Parameter Description

This parameter applies to area-scan camera operated in CTL mode, stating the maximum tolerated duration of the frame exposure duration as specified by the camera manufacturer.

It also applies to most the line-scan cameras, stating the maximum tolerated duration of the line exposure duration as specified by the camera manufacturer.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 microsecond *Minimum range value.*
20000000	20,000,000 microseconds (=20 seconds) *Maximum range value.*

FvalMode

Base DualBase Full FullXR

Usage of downstream signal FVAL

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1035 << 14	FvalMode	MC_FvalMode

Parameter Description

The Camera Link standard specifies a downstream signal aimed at signaling the validity of a video frame issued by the camera. This signal is called FVAL.

This parameter expresses the timing rules associated to FVAL.

Parameter Values

FN

Base DualBase Full FullXR

MC_FvalMode_FN
Description Frame None.

FA

Base DualBase Full FullXR

MC_FvalMode_FA
Description Frame Ante.

FC

Base DualBase Full FullXR

MC_FvalMode_FC
Description Frame Cover.

LvalMode

Base DualBase Full FullXR

Usage of downstream signal LVAL

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1036 << 14	LvalMode	MC_LvalMode

Parameter Description

The Camera Link standard specifies a downstream signal aimed at signaling the validity of a video line issued by the camera. This signal is called LVAL.

This parameter expresses the timing rules associated to LVAL.

Parameter Values

LA

Base DualBase Full FullXR

MC_LvalMode_LA
Description Line Ante.

LN

Base DualBase Full FullXR

MC_LvalMode_LN
Description Line None.

DvalMode

Base DualBase Full FullXR

Usage of downstream signal DVAL

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1037 << 14	DvalMode	MC_DvalMode

Parameter Description

This parameter expresses the timing rules associated to DVAL.

Parameter Values

DN

Base DualBase Full FullXR

MC_DvalMode_DN
Description Data None.

DG

Base DualBase Full FullXR

MC_DvalMode_DG
Description Data Gate.

CC1Usage

Base DualBase Full FullXR

Usage of upstream signal CC1

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1216 << 14	CC1Usage	MC_CC1Usage

Parameter Values

LOW

Base DualBase Full FullXR

MC_CC1Usage_LOW

Description
The control line is tied to the low logic state.

HIGH

Base DualBase Full FullXR

MC_CC1Usage_HIGH

Description
The control line is tied to the high logic state.

RESET

Base DualBase Full FullXR

MC_CC1Usage_RESET

Description
The control line implements the reset function.

AUXRESET

Base DualBase Full FullXR

MC_CC1Usage_AUXRESET

Description
The control line implements the auxiliary reset function.

SOFT

Base DualBase Full FullXR

MC_CC1Usage_SOFT

Description
The control line is controlled through the I/O API.

DIN1

Base DualBase Full FullXR

MC_CC1Usage_DIN1

Description
The control line is tied to the DIN1 input port.

IIN1

Base DualBase Full FullXR

MC_CC1Usage_IIN1

Description
The control line is tied to the IIN1 input port.

CC2Usage

Base DualBase Full FullXR

Usage of upstream signal CC2

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1218 << 14	CC2Usage	MC_CC2Usage

Parameter Values

LOW

Base DualBase Full FullXR

MC_CC2Usage_LOW

Description
The control line is tied to the low logic state.

HIGH

Base DualBase Full FullXR

MC_CC2Usage_HIGH

Description
The control line is tied to the high logic state.

RESET

Base DualBase Full FullXR

MC_CC2Usage_RESET

Description
The control line implements the reset function.

AUXRESET

Base DualBase Full FullXR

MC_CC2Usage_AUXRESET

Description
The control line implements the auxiliary reset function.

SOFT

Base DualBase Full FullXR

MC_CC2Usage_SOFT

Description
The control line is controlled through the I/O API.

DIN2

Base DualBase Full FullXR

MC_CC2Usage_DIN2

Description
The control line is tied to the DIN2 input port.

CC3Usage

Base DualBase Full FullXR

Usage of upstream signal CC3

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1219 << 14	CC3Usage	MC_CC3Usage

Parameter Values

LOW

Base DualBase Full FullXR

MC_CC3Usage_LOW

Description
The control line is tied to the low logic state.

HIGH

Base DualBase Full FullXR

MC_CC3Usage_HIGH

Description
The control line is tied to the high logic state.

RESET

Base DualBase Full FullXR

MC_CC3Usage_RESET

Description
The control line implements the reset function.

AUXRESET

Base DualBase Full FullXR

MC_CC3Usage_AUXRESET

Description
The control line implements the auxiliary reset function.

SOFT

Base DualBase Full FullXR

MC_CC3Usage_SOFT

Description
The control line is controlled through the I/O API.

IIN1

Base DualBase Full FullXR

MC_CC3Usage_IIN1

Description
The control line is tied to the IIN1 input port.

CC4Usage

Base DualBase Full FullXR

Usage of upstream signal CC4

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1220 << 14	CC4Usage	MC_CC4Usage

Parameter Values

LOW

Base DualBase Full FullXR

MC_CC4Usage_LOW

Description
The control line is tied to the low logic state.

HIGH

Base DualBase Full FullXR

MC_CC4Usage_HIGH

Description
The control line is tied to the high logic state.

RESET

Base DualBase Full FullXR

MC_CC4Usage_RESET

Description
The control line implements the reset function.

AUXRESET

Base DualBase Full FullXR

MC_CC4Usage_AUXRESET

Description
The control line implements the auxiliary reset function.

SOFT

Base DualBase Full FullXR

MC_CC4Usage_SOFT

Description
The control line is controlled through the I/O API.

TwoLineSynchronization

Base DualBase Full FullXR

Controls the two-line synchronization mode

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11049 << 14	TwoLineSynchronization	MC_TwoLineSynchronization

Parameter Description

This enumerated parameter controls the 2-line mode of the synchronized line-scan acquisition.

Parameter Usage

Relevance condition(s):

Condition: Basler Sprint Bilinear line-scan camera (or similar product).

Directive: Set this parameter to ENABLE to allow synchronized line-scan acquisition with bilinear linescan cameras such as the Basler Sprint.

Parameter Values

ENABLE

Base DualBase Full FullXR

MC_TwoLineSynchronization_ENABLE
Description The 2-line synchronization mode is enabled.

DISABLE

Base DualBase Full FullXR

MC_TwoLineSynchronization_DISABLE
Description The 2-line synchronization mode is disabled.
Default value.

TwoLineSynchronizationParity

Base DualBase Full FullXR

Controls the two-line synchronization parity

Parameter Info

Class	Category	Level	Type	Access
Channel	Camera Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11050 << 14	TwoLineSynchronizationParity	MC_TwoLineSynchronizationParity

Parameter Description

This enumerated parameter selects the line parity of individual cameras in a 2-line synchronized acquisition system.

Parameter Usage

Relevance condition(s):

Condition: TwoLineSynchronization = ENABLE

Parameter Values

ODD

Base DualBase Full FullXR

MC_TwoLineSynchronizationParity_ODD
Description The camera cycle begins at an odd line trigger count boundary.

EVEN

Base DualBase Full FullXR

MC_TwoLineSynchronizationParity_EVEN
Description The camera cycle begins at an even line trigger count boundary.
Default value.

4.4. Cable Features Category

Parameters setting the hardware attributes of the cable linking the camera to the frame grabber

ResetLine

Base DualBase Full FullXR

Designation of line chosen for transporting main reset to camera

Parameter Info

Class	Category	Level	Type	Access
Channel	Cable Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1158 << 14	ResetLine	MC_ResetLine

Parameter Description

This parameter declares which line is used inside the camera cable to connect the main reset signal from the frame grabber to the camera.

Some cameras use an additional reset control line to independently control the expose and read-out functions. Refer to the AuxResetLine parameter.

Parameter Values

CC1

Base DualBase Full FullXR

MC_ResetLine_CC1

Description
The main reset uses the CC1 camera control line.

CC2

Base DualBase Full FullXR

MC_ResetLine_CC2

Description
The main reset uses the CC2 camera control line.

CC3

Base DualBase Full FullXR

MC_ResetLine_CC3

Base DualBase Full FullXR

Description
The main reset uses the CC3 camera control line.

CC4

Base DualBase Full FullXR

MC_ResetLine_CC4

Description
The main reset uses the CC4 camera control line.

NC

Base DualBase Full FullXR

MC_ResetLine_NC

Description
The main reset is not used and not connected.

AuxResetLine

Base DualBase Full FullXR

Designation of line chosen for transporting auxiliary reset to camera

Parameter Info

Class	Category	Level	Type	Access
Channel	Cable Features	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1159 << 14	AuxResetLine	MC_AuxResetLine

Parameter Description

This parameter declares which line is used inside the camera cable to connect the auxiliary reset signal from the frame grabber to the camera.

Some cameras (area-scan or line-scan) use two reset control lines to independently control the expose and read-out functions. Refer to the ResetCtl parameter.

Parameter Values

NC

Base DualBase Full FullXR

MC_AuxResetLine_NC
Description No auxiliary reset line to connect.

CC1

Base DualBase Full FullXR

MC_AuxResetLine_CC1
Description The auxiliary reset uses the CC1 camera control line.

CC2

Base DualBase Full FullXR

MC_AuxResetLine_CC2
Description The auxiliary reset uses the CC2 camera control line.

CC3

Base DualBase Full FullXR

MC_AuxResetLine_CC3

Description
The auxiliary reset uses the CC3 camera control line.

CC4

Base DualBase Full FullXR

MC_AuxResetLine_CC4

Description
The auxiliary reset uses the CC4 camera control line.

4.5. Acquisition Control Category

Parameters installing the acquisition modes of the channel

AcquisitionMode

Fundamental acquisition mode

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	SELECT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
3396 << 14	AcquisitionMode	MC_AcquisitionMode

Parameter Description

Refer to the "MultiCam Acquisition Principles" on page 484 application note.

Parameter Values

WEB

Base DualBase Full FullXR

MC_AcquisitionMode_WEB

Description
This mode is intended for image acquisition of a continuous object, like a web, from a line-scan camera.
A single sequence acquiring SeqLength_Ln contiguous lines is available within the channel activity period. The sequence is divided in contiguous phases, each phase acquiring PageLength_Ln lines.
In the case SeqLength_Ln is not a multiple of PageLength_Ln , the surface is partially filled during the last phase.
The sequence and the first acquisition phase are initiated according to TrigMode . Subsequent acquisition phases are automatically initiated **without any line loss**.
BreakEffect specifies the behavior in case of a user break.

Default value.

PAGE

Base DualBase Full FullXR

MC_AcquisitionMode_PAGE

Description
This mode is intended for image acquisition of discrete objects from a line-scan camera.
Each page is constituted of contiguous lines; the page length, expressed in lines, is specified by PageLength_Ln .
A single sequence is capable to acquire SeqLength_Pg pages within the channel activity period.

LONGPAGE

Base DualBase Full FullXR

MC_AcquisitionMode_LONGPAGE

Description
This mode is intended for image acquisition of long or variable size discrete objects from a line-scan camera.
The parameter ActivityLength specifies the number of sequences within the channel activity period. Each sequence is capable to acquire SeqLength_Ln**contiguous** lines.
A sequence is divided in phases, each phase acquiring PageLength_Ln lines.

HFR

Base DualBase Full FullXR

MC_AcquisitionMode_HFR

Description
This mode is intended for acquisition of snapshot images from high frame rate area-scan cameras.
A single sequence is capable to acquire SeqLength_Fr frames within the channel activity period. The sequence is divided into phases, each phase acquiring PhaseLength_Fr frames into a single destination surface.

SNAPSHOT

Base DualBase Full FullXR

MC_AcquisitionMode_SNAPSHOT

Description
This mode is intended for acquisition of snapshot images from area-scan cameras.
The unique sequence is capable to acquire SeqLength_Fr frames within the channel activity period.

SynchronizedAcquisition

Base DualBase Full FullXR

Inter-Channel synchronization mode

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	SELECT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10571 << 14	SynchronizedAcquisition	MC_SynchronizedAcquisition

Parameter Description

Main control of the inter-channel synchronization though the SyncBus.

Parameter Usage

Directive: Set to MASTER for one SyncBus contributor and set to SLAVE for all other contributors.

Directive: Alternatively, when all contributors belong to the same card, set to LOCAL_MASTER for one SyncBus contributor and set to SLAVE for all other contributors.

Directive: Set to LOCAL_SLAVE for only one SyncBus contributor when all contributors belong to the same card.

Parameter Values

OFF

Base DualBase Full FullXR

MC_SynchronizedAcquisition_OFF
Description The inter-channel synchronized acquisition feature is disabled. The MultiCam channel is operating independently from other MultiCam channels.
Default value.

MASTER

Base DualBase Full FullXR

MC_SynchronizedAcquisition_MASTER

Description
The MultiCam channel is configured as the SyncBus master agent. Two synchronization signals are delivered on the IOUT3 and IOUT4 output ports of the channel for distribution to all the SyncBus agents using the appropriate wiring. The acquisition controller gets synchronization signals from the SyncBus through the IIN3 and IIN4 input ports of the channel.

SLAVE

Base DualBase Full FullXR

MC_SynchronizedAcquisition_SLAVE

Description
The MultiCam channel is configured as a SyncBus slave agent. The acquisition controller gets synchronization signals from the SyncBus through the IIN3 and IIN4 input ports of the channel.

LOCAL_MASTER

DualBase

MC_SynchronizedAcquisition_LOCAL_MASTER

Description
The MultiCam channel is configured as the local SyncBus master agent. Two synchronization signals are delivered on a local SyncBus for distribution to all the local SyncBus agents using an internal wiring. The acquisition controller gets synchronization signals from the local SyncBus.

LOCAL_SLAVE

Base DualBase Full FullXR

MC_SynchronizedAcquisition_LOCAL_SLAVE

Description
The MultiCam channel is configured as a local SyncBus master agent. The acquisition controller gets synchronization signals from the local SyncBus.

SynchronizedAcquisitionBus

Full FullXR

SyncBus interface selector

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	SELECT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11217 << 14	SynchronizedAcquisitionBus	MC_SynchronizedAcquisitionBus

Parameter Description

Selects the hardware interface used by the SyncBus.

Parameter Usage

Directive: Set to C2C when the line rate exceeds 40 kHz.

Parameter Values

ISO

Full FullXR

MC_SynchronizedAcquisitionBus_ISO
Description The SyncBus uses the IIN3/IIN4 isolated input lines and the IOUT3/IOUT4 isolated output lines.
Default value.

C2C

Full FullXR

MC_SynchronizedAcquisitionBus_C2C
Description The SyncBus uses the C2C SyncBus connector.

SynchronizedPageTrigger

Full FullXR

Page trigger synchronization control

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	SELECT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11221 << 14	SynchronizedPageTrigger	MC_SynchronizedPageTrigger

Parameter Description

Selects the signal on which page triggers are synchronized before being broadcasted on the SyncBus.

Parameter Usage

Relevance condition(s):

Condition: BoardTopology = MONO_DECA

Condition: Synchronized acquisition using SyncBus

Condition: Line-scan camera

Directive: The LINETRIGGER default value is only applicable to line-scan cameras controlled by the frame grabber (RC, RG or RP camera control methods).

Directive: Setting to LVALRISE allows to share page triggers using the SyncBus when the camera is not controlled by the frame grabber (SC cameral control method).

Parameter Values

LINETRIGGER

Full FullXR

MC_SynchronizedPageTrigger_LINETRIGGER
Description Page triggers are synchronized with the next line trigger event.
Default value.

LVALRISE

Full FullXR

MC_SynchronizedPageTrigger_LVALRISE

Description
Page triggers are synchronized with the next start of line event (LVAL rising edge).

PageCaptureMode

Full FullXR

Start-of-page capture control

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	SELECT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11222 << 14	PageCaptureMode	MC_PageCaptureMode

Parameter Description

This parameter controls the conditions applied by the frame grabber to capture the first data line after a page trigger.

Parameter Usage

Relevance condition(s):

Condition: BoardTopology = MONO_DECA

Condition: Line-scan camera

Parameter Values

FIRST_LINE

Full FullXR

MC_PageCaptureMode_FIRST_LINE
Description The first captured line is the first entire data line sent by the camera after the page trigger event.
Default value.

FIRST_EXPOSURE

Full FullXR

MC_PageCaptureMode_FIRST_EXPOSURE
Description The first captured line is the data line resulting from the first entire exposure cycle after the page trigger event.

TrigMode

Grabber acquisition sequence triggering mode

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
512 << 14	TrigMode	MC_TrigMode

Parameter Description

The TrigMode parameter establishes the starting conditions of an acquisition sequence.

Refer to the "MultiCam Acquisition Principles" on page 484 application note.

Parameter Values

IMMEDIATE

Base DualBase Full FullXR

MC_TrigMode_IMMEDIATE
Description The acquisition sequence starts immediately without waiting for a trigger.

HARD

Base DualBase Full FullXR

MC_TrigMode_HARD
Description The start of the acquisition sequence is delayed until the hardware trigger line senses a valid transition. Parameters TrigLine or TrigLineIndex specify the location of a hardware trigger input line. Parameters TrigCtl , TrigEdge and TrigFilter specify the configuration of the hardware trigger input line. A programmable delay can be inserted with parameter PageDelay_Ln.

SOFT

Base DualBase Full FullXR

MC_TrigMode_SOFT

Description
The start of the acquisition sequence is delayed until the software sets parameter ForceTrig to TRIG.

COMBINED

Base DualBase Full FullXR

MC_TrigMode_COMBINED

Description
The start of the acquisition sequence is delayed until detection of hardware or software trigger.

SLAVE

Base DualBase Full FullXR

MC_TrigMode_SLAVE

Description
The start of the acquisition sequence is originated from the master device.

Applicability condition(s)
Condition: SynchronizedAcquisition is set to SLAVE or LOCAL_SLAVE.

NextTrigMode

Grabber subsequent acquisition phases or slices triggering mode

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
663 << 14	NextTrigMode	MC_NextTrigMode

Parameter Description

This parameter establishes the starting conditions of the subsequent acquisition phases or slices.

Refer to the "MultiCam Acquisition Principles" on page 484 application note.

On Domino boards, the default value is SAME.

On Grablink boards, the default value depends on the selected AcquisitionMode :

- When WEB or LONGPAGE, the default value is REPEAT.
- When SNAPSHOT, HFR or PAGE, the default value is SAME.

Parameter Values

COMBINED

Base DualBase Full FullXR

MC_NextTrigMode_COMBINED

Description
Any subsequent acquisition phase or slice is delayed until detection of hardware or software trigger.

HARD

Base DualBase Full FullXR

MC_NextTrigMode_HARD

Description
Any subsequent acquisition phase or slice is delayed until the hardware trigger line senses a valid transition.
Parameters TrigLine or TrigLineIndex specifies the location of a hardware trigger input line.
Parameters TrigCtl , TrigEdge and TrigFilter specify the configuration of the hardware trigger input line.
A programmable delay can be inserted with parameter PageDelay_Ln.

REPEAT

Base DualBase Full FullXR

MC_NextTrigMode_REPEAT

Description
Any subsequent acquisition phase or slice occurs immediately after the preceding one.

SAME

Base DualBase Full FullXR

MC_NextTrigMode_SAME

Description
Any subsequent acquisition phase or slice occurs similarly to the conditions defined by TrigMode.

SOFT

Base DualBase Full FullXR

MC_NextTrigMode_SOFT

Description
Any subsequent acquisition phase or slice is delayed until the software sets parameter ForceTrig to TRIG.

SLAVE

Base DualBase Full FullXR

MC_NextTrigMode_SLAVE

Description
Any subsequent acquisition phase or slice is delayed until a trigger is originated from the master device.

Applicability condition(s)
Condition: SynchronizedAcquisition is set to SLAVE or LOCAL_SLAVE.

TrigRepeatCount

Base DualBase Full FullXR

Trigger repetition control

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
10666 << 14	TrigRepeatCount	MC_TrigRepeatCount

Parameter Description

This parameter controls the trigger repetition, a feature providing the capability to insert additional acquisition phases after each triggered acquisition phase.

A value of 0 disables the trigger repetition feature. A positive value enables the trigger repetition feature and specifies the number of additional acquisition phases inserted after every triggered phase.

Parameter Usage

Prerequisite action(s):

Condition: AcquisitionMode = SNAPSHOT, HFR or PAGE.

Condition: The feature applies on triggered acquisition phases only. Refer to "TrigMode" on page 211 and "NextTrigMode" on page 213.

Directive: Trigger overlap is allowed during the last repeated acquisition phase but not before!

Parameter Values

Value	Description
0	*Minimum range value. Default value.*
1024	*Maximum range value.*

EndTrigMode

Grabber end triggering mode

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
2916 << 14	EndTrigMode	MC_EndTrigMode

Parameter Description

The EndTrigMode parameter establishes the conditions of a sequence termination.

Refer to the "MultiCam Acquisition Principles" on page 484 application note.

Parameter Values

AUTO

Base DualBase Full FullXR

MC_EndTrigMode_AUTO

Description
The acquisition sequence terminates automatically upon expiration of a frame, page or line counter. See **Automatic completion conditions vs. AcquisitionMode** below.

HARD

Base DualBase Full FullXR

MC_EndTrigMode_HARD

Description
The acquisition sequence terminates upon the detection of a valid transition of the hardware end-trigger line.
Parameters EndTrigCtl , EndTrigEdge , EndTrigFilter and EndTrigLine specify the location and the configuration of the hardware end-trigger input line.
A programmable delay can be inserted with parameter EndPageDelay_Ln .

Applicability condition(s)
Condition: AcquisitionMode is set to VIDEO or LONGPAGE.
Base DualBase Full FullXR
Condition: AcquisitionMode is set to LONGPAGE.

SLAVE

Base DualBase Full FullXR

MC_EndTrigMode_SLAVE

Description
The end of the acquisition sequence is originated from the master device.

Applicability condition(s)
Condition: AcquisitionMode is set to LONGPAGE andSynchronizedAcquisition is set to SLAVE or LOCAL_SLAVE.

BreakEffect

Grabber break effect on the acquisition phase

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
2011 << 14	BreakEffect	MC_BreakEffect

Parameter Description

The BreakEffect parameter establishes the effect of a user break on the channel.

Refer to the "MultiCam Acquisition Principles" on page 484 application note.

Parameter Values

FINISH

Base DualBase Full FullXR

MC_BreakEffect_FINISH
Description The effect of the user break is postponed until the acquisition sequence reaches a specific boundary. The effect is immediate only when no acquisition has been triggered.
Applicability condition(s) *Condition:* EndTrigMode is set to AUTO *Condition:* AcquisitionMode is set to VIDEO, SNAPSHOT, or HFR: the channel activity and the sequence terminate at a frame boundary. *Condition:* AcquisitionMode is set to WEB, PAGE, or LONGPAGE: the channel activity and the sequence terminate at a page boundary.

ABORT

Base DualBase Full FullXR

MC_BreakEffect_ABORT
Description The effect of the user break is immediate. The current acquisition is incomplete. The portion of image already acquired is available. This value is only available for line-scan acquisition modes, not for HFR.

ActivityLength

Acquisition sequences count

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
3406 << 14	ActivityLength	MC_ActivityLength

Parameter Description

An activity period of a channel is made of one or several acquisition sequences. This parameter establishes the number of acquisition sequences constituting a channel activity period.

MultiCam sets this parameter to 1 when AcquisitionMode is SNAPSHOT, WEB, PAGE or HFR.

Setting ActivityLength to MC_INDETERMINATE results in indefinitely repeated acquisition sequences. A user break is required to stop the channel activity.

Refer to the "MultiCam Acquisition Principles" on page 484 application note.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 acquisition sequence *Condition:* AcquisitionMode = SNAPSHOT or HFR or WEB or PAGE
MC_INDETERMINATE	Undefined number of acquisition sequences *Condition:* AcquisitionMode = LONGPAGE

PageLength_Ln

Base DualBase Full FullXR

Length of page acquisition, expressed as a number of lines

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
1039 << 14	PageLength_Ln	MC_PageLength_Ln

Parameter Description

This parameter is applicable to line-scan acquisition, and declares the number of scanned lines stored into a surface.

The user is invited to set this parameter when AcquisitionMode = PAGE, WEB or LONGPAGE.

The user is invited to read back the parameter since MultiCam may trim its value to fulfill specific grabber requirements.

Refer to the "MultiCam Acquisition Principles" on page 484 application note.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 line *Minimum range value.*
65535	65,535 lines *Maximum range value.*

SeqLength_Fr

Number of frames in a sequence

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
3407 << 14	SeqLength_Fr	MC_SeqLength_Fr

Parameter Description

This parameter establishes the number of frames constituting a sequence.

The user is invited to set this parameter when EndTrigMode is AUTO and AcquisitionMode has one of the following values: VIDEO, SNAPSHOT or HFR.

Refer to the "MultiCam Acquisition Principles" on page 484 application note.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 frame *Condition:* AcquisitionMode is set to SNAPSHOT or HFR *Minimum range value.*
65534	65,534 frames *Condition:* AcquisitionMode is set to SNAPSHOT *Maximum range value.*
16711170	16,711,170 frames for max. value PhaseLenght_Fr otherwise (PhaseLength_Fr × 65,534) *Condition:* AcquisitionMode is set to HFR *Maximum range value.*
MC_INDETERMINATE	The frame acquisition is repeated indefinitely, a user break is required to terminate a sequence

SeqLength_Pg

Base DualBase Full FullXR

Number of pages in a sequence

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
3408 << 14	SeqLength_Pg	MC_SeqLength_Pg

Parameter Description

The SeqLength_Pg parameter establishes the number of pages constituting a sequence.

The user is invited to set this parameter when EndTrigMode is AUTO and AcquisitionMode is PAGE.

Setting SeqLength_Pg to MC_INDETERMINATE results in indefinitely repeated pages acquisition. A user break is required to terminate the sequence.

Refer to the "MultiCam Acquisition Principles" on page 484 application note.

Parameter Usage

Relevance condition(s):

Condition: AcquisitionMode is set to PAGE

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 page *Minimum range value.*
65534	65,534 pages *Maximum range value.*
MC_INDETERMINATE	The page acquisition is repeated indefinitely, a user break is required to terminate a sequence

SeqLength_Ln

Base DualBase Full FullXR

Number of acquired lines in a sequence

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
3401 << 14	SeqLength_Ln	MC_SeqLength_Ln

Parameter Description

The SeqLength_Ln parameter establishes the number of lines to be acquired in a sequence.

The user is invited to set this parameter when EndTrigMode is AUTO and AcquisitionMode has one of the following values: WEB or LONGPAGE.

Setting SeqLength_Ln to MC_INDETERMINATE results in indefinitely repeated page acquisition. A user break is required to terminate the sequence.

Refer to the "MultiCam Acquisition Principles" on page 484 application note.

Parameter Usage

Relevance condition(s):

Condition: AcquisitionMode is set to WEB or LONGPAGE

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 line *Minimum range value.*
Variable	(PageLength_Ln * 65,535) frames *Maximum range value.*
MC_INDETERMINATE	The line acquisition is repeated indefinitely, a user break is required to terminate a sequence *Condition:* AcquisitionMode is set to WEB

SeqLength_Ph

Number of acquisition phases constituting a sequence

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	ADJUST	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
3399 << 14	SeqLength_Ph	MC_SeqLength_Ph

Parameter Description

The user is invited to get the value of this parameter when EndTrigMode is set to AUTO.

Refer to the "MultiCam Acquisition Principles" on page 484 application note.

PhaseLength_Fr

Number of frames constituting a phase

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
3409 << 14	PhaseLength_Fr	MC_PhaseLength_Fr

Parameter Description

The parameter establishes the total number of frames acquired within an acquisition phase.

Refer to the "MultiCam Acquisition Principles" on page 484 application note.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 frame *Condition:* AcquisitionMode is set to SNAPSHOT.
1	1 frame *Condition:* AcquisitionMode is set to HFR. *Minimum range value.*
255	255 frames *Condition:* AcquisitionMode is set to HFR. *Maximum range value.*

PhaseLength_Pg

Base DualBase Full FullXR

Number of pages constituting a phase

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
3418 << 14	PhaseLength_Pg	MC_PhaseLength_Pg

Parameter Description

This parameter establishes the total number of pages acquired within an acquisition phase.

Refer to the "MultiCam Acquisition Principles" on page 484 application note.

Elapsed_Fr

Elapsed number of acquired frames

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
3453 << 14	Elapsed_Fr	MC_Elapsed_Fr

Parameter Description

This parameter gives information about the acquisition sequence progress, by reporting the number of completed frame acquisitions in a sequence.

Remaining_Fr

Number of remaining frames to acquire

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
3454 << 14	Remaining_Fr	MC_Remaining_Fr

Parameter Description

This parameter gives information about the acquisition sequence progress, by reporting the number of remaining frames to acquire in a sequence.

Parameter Usage

Relevance condition(s):

Condition: Seqlength_Fr is not set to MC_INDETERMINATE

Parameter Values

Value	Description
0	*Minimum range value.*

PerSecond_Fr

Number of frames acquired during a second

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
3452 << 14	PerSecond_Fr	MC_PerSecond_Fr

Elapsed_Pg

Base DualBase Full FullXR

Elapsed number of acquired pages

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
3455 << 14	Elapsed_Pg	MC_Elapsed_Pg

Parameter Description

This parameter gives information about the acquisition sequence progress, by reporting the number of completed page acquisitions in a sequence.

Remaining_Pg

Base DualBase Full FullXR

Number of remaining pages to acquire

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
3457 << 14	Remaining_Pg	MC_Remaining_Pg

Parameter Description

This parameter gives information about the acquisition sequence progress, by reporting the number of remaining pages to acquire in a sequence.

Parameter Usage

Relevance condition(s):

Condition: Seqlength_Pg is not set to MC_INDETERMINATE

Parameter Values

Base DualBase Full FullXR

Value	Description
0	*Minimum range value.*

Elapsed_Ln

Base DualBase Full FullXR

Elapsed number of acquired lines

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
3456 << 14	Elapsed_Ln	MC_Elapsed_Ln

Parameter Description

This parameter gives information about the acquisition sequence progress, by reporting the number of completed line acquisitions in a sequence.

Remaining_Ln

Base DualBase Full FullXR

Number of remaining lines to acquire

Parameter Info

Class	Category	Level	Type	Access
Channel	Acquisition Control	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
3458 << 14	Remaining_Ln	MC_Remaining_Ln

Parameter Description

This parameter gives information about the acquisition sequence progress, by reporting the number of remaining lines to acquire in a sequence.

Parameter Usage

Relevance condition(s):

Condition: Seqlength_Ln is not set to MC_INDETERMINATE

Parameter Values

Base DualBase Full FullXR

Value	Description
0	*Minimum range value.*

4.6. Trigger Control Category

Parameters controlling the triggering features associated to the channel

TrigCtl

Base DualBase Full FullXR

Electrical style of the trigger hardware line

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
513 << 14	TrigCtl	MC_TrigCtl

Parameter Description

This parameter specifies the electrical style of the GPIO line used as trigger input.

Along with TrigEdge and TrigFilter, it declares the grabber attributes of the trigger line sensed by the channel and aimed at generating the trigger event.

Parameter Usage

Relevance condition(s):

Condition: A hardware line is used as trigger input. TrigMode or NextTrigMode are set to HARD or COMBINED.

Parameter Values

DIFF

Base DualBase Full FullXR

MC_TrigCtl_DIFF
Description Differential high-speed input compatible with EIA/TIA-422 signaling.

ISO

Base DualBase Full FullXR

MC_TrigCtl_ISO
Description Isolated current loop input compatible with TTL, +12V, +24V signaling.
Default value.

CAMERA

Base DualBase Full FullXR

MC_TrigCtl_CAMERA

Description
Camera Link downstream signaling.

Applicability condition(s)
Condition: BoardTopology ≠ MONO_SLOW
Condition: BoardTopology ≠ DUO_SLOW

TrigEdge

Base DualBase Full FullXR

Significant edge of designated trigger hardware line from outside system

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
664 << 14	TrigEdge	MC_TrigEdge

Parameter Description

This parameter applies to the hardware line designated by TrigLine or TrigLineIndex .

Along with TrigCtl and TrigFilter , it declares the grabber attributes of the trigger line sensed by the channel and aimed at generating the trigger event.

Parameter Values

GOHIGH

Base DualBase Full FullXR

MC_TrigEdge_GOHIGH
Description The trigger event is generated at each positive-going transition of the trigger line.

GOLOW

Base DualBase Full FullXR

MC_TrigEdge_GOLOW
Description The trigger event is generated at each negative-going transition of the trigger line.

TrigFilter

Base DualBase Full FullXR

Noise removal on designated trigger hardware line from outside system

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
665 << 14	TrigFilter	MC_TrigFilter

Parameter Description

This parameter applies to the hardware line designated by TrigLine or TrigLineIndex .

Along with TrigCtl and TrigEdge , it declares the grabber attributes of the trigger line sensed by the channel and aimed at generating the trigger event.

The TrigFilter parameter specifies the time constant of the noise reduction filter of the designated hardware line.

The time constant of the filter is the amount of time the line should be detected at the same logic state before a logic transition be considered.

When insulated I/Os are used, TrigFilter is MEDIUM or STRONG. The value OFF is not allowed.

Parameter Values

OFF

Base DualBase Full FullXR

MC_TrigFilter_OFF
Description The noise removal filter is turned off.
Base DualBase Full FullXR *Description* Time constant = 100 ns

ON

Base DualBase Full FullXR

MC_TrigFilter_ON

Description
The noise removal filter is turned on.

Base DualBase Full FullXR

Description
Time constant = 500 ns

MEDIUM

Base DualBase Full FullXR

MC_TrigFilter_MEDIUM

Description
The noise removal filter is turned on with a moderate filtering effect.

Base DualBase Full FullXR

Description
Time constant = 500 ns

STRONG

Base DualBase Full FullXR

MC_TrigFilter_STRONG

Description
The noise removal filter is turned on with a strong filtering effect.

Base DualBase Full FullXR

Description
Time constant = 2500 ns

TrigDelay_us

Base DualBase Full FullXR

Trigger delay before the reset pulse is sent to the camera, expressed in microseconds

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
2153 << 14	TrigDelay_us	MC_TrigDelay_us

Parameter Description

This parameter can be used to insert a delay between the hardware trigger and the reset pulse sent to the camera.

NOTE
This parameter does not affect software triggers.

NOTE
This parameter is applicable exclusively for area-scan cameras. For line-scan cameras, use instead PageDelay_Ln .

Parameter Values

Base DualBase Full FullXR

Value	Description
0	No delay *Minimum range value.*
2000000	2,000,000 microseconds (= 2 seconds) *Maximum range value.*

PageDelay_Ln

Base DualBase Full FullXR

Delay from trigger to start the page acquisition, expressed as a number of scanned lines

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
1038 << 14	PageDelay_Ln	MC_PageDelay_Ln

Parameter Description

This parameter can be used to insert a programmable delay between the hardware trigger and the start of the acquisition.

It is expressed as a number of scanned lines. It exclusively applies to line-scan cameras when AcquisitionMode is LONGPAGE or PAGE.

The waiting phase corresponding to the countdown of the page delay can overlap the previous page acquisition phase.

NOTE
For area-scan cameras, use TrigDelay_us instead.

Parameter Usage

Directive: Use this feature to compensate the delay introduced by a position detector placed away from the camera field of view.

Parameter Values

Base DualBase Full FullXR

Value	Description
0	No delay *Minimum range value.*
65534	65,534 lines delay *Maximum range value.*

TrigDelay_Pls

Base DualBase Full FullXR

Number of hardware trigger pulses to ignore after the start of sequence event

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
10080 << 14	TrigDelay_Pls	MC_TrigDelay_Pls

Parameter Description

This parameter specifies the number of detected pulses on the hardware trigger line to be skipped after the acquisition sequence begins.

This parameter applies when acquisition control settings require a hardware trigger or page trigger (all acquisition modes).

Parameter Values

Base DualBase Full FullXR

Value	Description
0	No delay *Minimum range value.*
65536	65,536 ignored pulses *Maximum range value.*

NextTrigDelay_Pls

Base DualBase Full FullXR

Number of hardware trigger pulses to skip between successive acquisition phases

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
10081 << 14	NextTrigDelay_Pls	MC_NextTrigDelay_Pls

Parameter Description

This parameter specifies the number of detected pulses on the hardware trigger line to be skipped between successive acquisition phases.

This parameter applies when acquisition control settings require a hardware trigger or page trigger event for subsequent acquisition phases (SNAPSHOT, HFR and PAGE acquisition modes)

Parameter Values

Base DualBase Full FullXR

Value	Description
0	No delay *Minimum range value. Default value.*
65536	65,536 skipped pulses *Maximum range value.*

EndTrigCtl

Base DualBase Full FullXR

Electrical style of designated end trigger hardware line

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
3447 << 14	EndTrigCtl	MC_EndTrigCtl

Parameter Description

This parameter specifies the electrical style of the GPIO line used as end trigger input.

Along with EndTrigEdge and EndTrigFilter, it declares the grabber attributes of the trigger line sensed by the channel and aimed at generating the end trigger event.

Parameter Usage

Prerequisite action(s):

Condition: AcquisitionMode = LONGPAGE

Condition: EndTrigMode = HARD

Parameter Values

DIFF

Base DualBase Full FullXR

MC_EndTrigCtl_DIFF
Description Differential high-speed input compatible with EIA/TIA-422 signaling.

ISO

Base DualBase Full FullXR

MC_EndTrigCtl_ISO
Description Isolated current loop input compatible with TTL, +12V, +24V signaling.
Default value.

CAMERA

Base DualBase Full FullXR

MC_EndTrigCtl_CAMERA

Description
Isolated current loop input compatible with TTL, +12V, +24V signaling.

Default value.

Applicability condition(s)
Condition: BoardTopology ≠ MONO_SLOW
Condition: BoardTopology ≠ DUO_SLOW

EndTrigEdge

Base DualBase Full FullXR

Significant edge of designated end trigger hardware line from outside system

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
2905 << 14	EndTrigEdge	MC_EndTrigEdge

Parameter Description

This parameter applies to the hardware line designated by EndTrigLine or EndTrigLineIndex .

Along with EndTrigCtl and EndTrigFilter , it declares the grabber attributes of the end trigger line sensed by the channel and aimed at generating the end trigger event.

EndTrigEdge determines the significant edge of the end trigger pulse.

Parameter Values

GOHIGH

Base DualBase Full FullXR

MC_EndTrigEdge_GOHIGH
Description The trigger event is generated at each positive-going transition of the trigger line.

GOLOW

Base DualBase Full FullXR

MC_EndTrigEdge_GOLOW
Description The trigger event is generated at each negative-going transition of the trigger line.

EndTrigFilter

Base DualBase Full FullXR

Noise removal on designated end trigger hardware line from outside system

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
3639 << 14	EndTrigFilter	MC_EndTrigFilter

Parameter Description

This parameter applies to the hardware line designated by EndTrigLine or EndTrigLineIndex .

Along with EndTrigCtl , EndTrigFilter declares the grabber attributes of the end trigger line sensed by the channel and aimed at generating the end trigger event.

EndTrigFilter specifies the time constant of the noise reduction filter of the designated hardware line.

The time constant of the filter is the amount of time the line should be detected at the same logic state before a logic transition be considered.

When insulated I/O are used, EndTrigFilter is MEDIUM or STRONG. The value OFF is not allowed.

Parameter Values

OFF

Base DualBase Full FullXR

MC_EndTrigFilter_OFF

Base DualBase Full FullXR

Description
The filter time constant is approximately 100 ns.

Applicability condition(s)
Condition: This value is not allowed when EndTrigFilter is set to RELAY.

ON

Base DualBase Full FullXR

MC_EndTrigFilter_ON

Description
The filter time constant is approximately 500 ns.

MEDIUM

Base DualBase Full FullXR

MC_EndTrigFilter_MEDIUM

Description
The filter time constant is approximately 500 ns.

STRONG

Base DualBase Full FullXR

MC_EndTrigFilter_STRONG

Description
The filter time constant is approximately 2500 ns.

Default value.

EndTrigEffect

Base DualBase Full FullXR

Effect of the "End Trigger" event

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10672 << 14	EndTrigEffect	MC_EndTrigEffect

Parameter Description

Selects the effect of an "End Trigger" event on the end of the acquisition phase.

Parameter Usage

Relevance condition(s):

Condition: AcquisitionMode = LONGPAGE

Condition: EndTrigMode = HARD

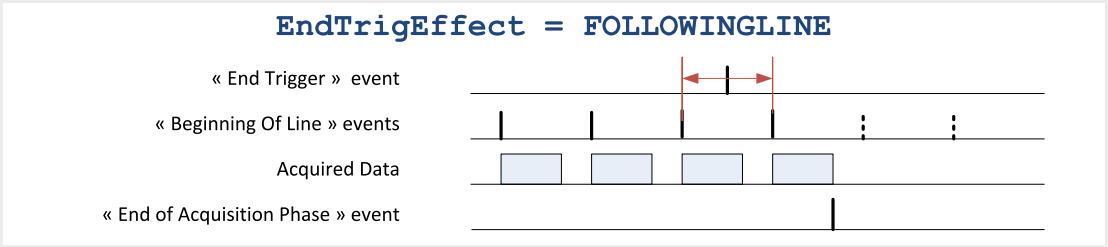
Parameter Values

FOLLOWINGLINE

Base DualBase Full FullXR

MC_EndTrigEffect_FOLLOWINGLINE

Description
On reception of an "End Trigger" event, the MultiCam Acquisition Controller acquires the line following the "End Trigger" event then terminates the acquisition phase.



Terminates after the following line

Default value.

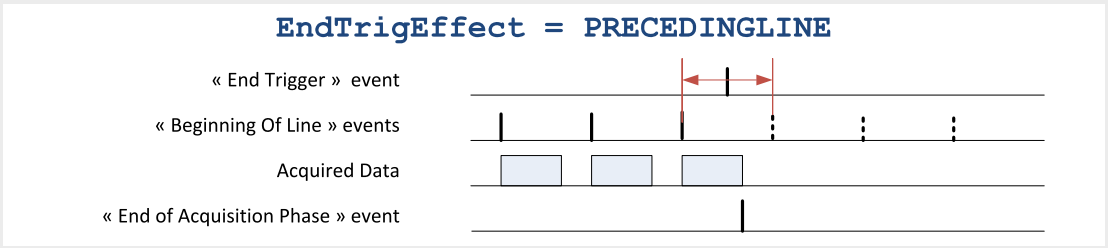
PRECEDINGLINE

Base DualBase Full FullXR

MC_EndTrigEffect_PRECEDINGLINE

Description

On reception of an "End Trigger" event, the MultiCam Acquisition Controller acquires the line preceding the "End Trigger" event and terminates the acquisition phase immediately.



NOTE

The PRECEDINGLINE value is not allowed for Bayer bi-linear line-scan cameras.

EndPageDelay_Ln

Base DualBase Full FullXR

Delay from end trigger to end of page acquisition, expressed as a number of scanned lines

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
3723 << 14	EndPageDelay_Ln	MC_EndPageDelay_Ln

Parameter Description

This parameter can be used to insert a programmable delay between the hardware end trigger and the end of the acquisition.

It is expressed as a number of scanned lines. It exclusively applies to line-scan cameras when AcquisitionMode is LONGPAGE.

Parameter Usage

Directive: Use this feature to compensate the delay introduced by a position detector placed away from the camera field of view.

Parameter Values

Base DualBase Full FullXR

Value	Description
0	No delay *Minimum range value.*
65534	65534 lines delay *Maximum range value.*

ForceTrig

Forces an event trigger from the application

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	ADJUST	Enumerated	Set Only

Num ID	String Identifier	C, C++ identifier
50 << 14	ForceTrig	MC_ForceTrig

Parameter Description

Refer to the "MultiCam Acquisition Principles" on page 484 application note.

Parameter Values

TRIG

MC_ForceTrig_TRIG
Description Forces a soft trigger event.

ENDTRIG

Base DualBase Full FullXR

MC_ForceTrig_ENDTRIG
Description Forces a soft end trigger event.
Applicability condition(s) *Condition:* AcquisitionMode is set to LONGPAGE.

TrigLine

Base DualBase Full FullXR

Designation of the trigger hardware line

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
666 << 14	TrigLine	MC_TrigLine

Parameter Description

This parameter designates the GPIO line sensed by the channel and aimed at generating the trigger event.

Along with "TrigCtl" on page 235, "TrigEdge" on page 237 and "TrigFilter" on page 238, it declares the grabber attributes of the trigger line sensed by the channel and aimed at generating the trigger event.

Parameter Usage

Relevance condition(s):

Condition: A hardware line is used as trigger input.TrigMode or NextTrigMode are set to HARD or COMBINED.

Prerequisite action(s):

Condition: TrigCtl is set according to the desired electrical style.

Parameter Values

NOM

Base DualBase Full FullXR

MC_TrigLine_NOM

Base DualBase Full FullXR

Description

Selects the nominal line corresponding to the pre-selected TrigCtl value:

- DIN2 when TrigCtl = DIFF
- IIN2 when TrigCtl = ISO
- FVAL when TrigCtl = CAMERA

Default value.

DIN1

Base DualBase Full FullXR

MC_TrigLine_DIN1

Description
Differential high-speed input lines pair #1.

Applicability condition(s)
Condition: TrigCtl = DIFF

DIN2

Base DualBase Full FullXR

MC_TrigLine_DIN2

Description
Differential high-speed input lines pair #2.

Applicability condition(s)
Condition: TrigCtl = DIFF

IIN1

Base DualBase Full FullXR

MC_TrigLine_IIN1

Description
Isolated current loop input line #1.

Applicability condition(s)
Condition: TrigCtl = ISO

IIN2

Base DualBase Full FullXR

MC_TrigLine_IIN2

Description
Isolated current loop input line #2.

Applicability condition(s)
Condition: TrigCtl = ISO

IIN3

Base DualBase Full FullXR

MC_TrigLine_IIN3

Description
Isolated current loop input line #3.

Applicability condition(s)
Condition: TrigCtl = ISO

IIN4

Base DualBase Full FullXR

MC_TrigLine_IIN4
Description Isolated current loop input line #4.
Applicability condition(s) *Condition:* TrigCtl = ISO

EndTrigLine

Base DualBase Full FullXR

Designation of the end trigger hardware line

Parameter Info

Class	Category	Level	Type	Access
Channel	Trigger Control	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
2906 << 14	EndTrigLine	MC_EndTrigLine

Parameter Description

This parameter designates the GPIO line sensed by the channel and aimed at generating the end trigger event.

Along with "EndTrigCtl" on page 244 and "EndTrigFilter" on page 247, it declares the grabber attributes of the trigger line sensed by the channel and aimed at generating the end trigger event.

Parameter Usage

Prerequisite action(s):

Condition: AcquisitionMode = LONGPAGE

Condition: EndTrigMode = HARD

Parameter Values

NOM

Base DualBase Full FullXR

MC_EndTrigLine_NOM
Description
Base DualBase Full FullXR *Description* Selects the nominal line corresponding to the pre-selected EndTrigCtl value: • DIN2 when EndTrigCtl = DIFF • IIN2 when EndTrigCtl = ISO • FVAL when TrigCtl = CAMERA
Default value.

DIN1

Base DualBase Full FullXR

MC_EndTrigLine_DIN1

Description
Differential high-speed input lines pair #1.

Applicability condition(s)
Condition: EndTrigCtl = DIFF

DIN2

Base DualBase Full FullXR

MC_EndTrigLine_DIN2

Description
Differential high-speed input lines pair #2.

Applicability condition(s)
Condition: EndTrigCtl = DIFF

IIN1

Base DualBase Full FullXR

MC_EndTrigLine_IIN1

Description
Isolated current loop input line #1.

Applicability condition(s)
Condition: EndTrigCtl = ISO

IIN2

Base DualBase Full FullXR

MC_EndTrigLine_IIN2

Description
Isolated current loop input line #2.

Applicability condition(s)
Condition: EndTrigCtl = ISO

IIN3

Base DualBase Full FullXR

MC_EndTrigLine_IIN3

Description
Isolated current loop input line #3.

Applicability condition(s)
Condition: EndTrigCtl = ISO

IIN4

Base DualBase Full FullXR

MC_EndTrigLine_IIN4

Description
Isolated current loop input line #4.

Applicability condition(s)
Condition: EndTrigCtl = ISO

4.7. Interleaved Acquisition Category

Parameters controlling the interleaved acquisition feature

InterleavedAcquisition

Base DualBase Full FullXR

Master control switch of the interleaved acquisition feature

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	SELECT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10936 << 14	InterleavedAcquisition	MC_InterleavedAcquisition

Parameter Description

This parameter allows to switch ON or OFF the interleaved acquisition feature.

When Interleaved Acquisition is turned ON, the Camera and Illumination Controller is configured with two different programs named P1 and P2. The programs are executed alternatively, starting with P1.

For more information, refer to the Interleaved Acquisition section of the Grablink Documentation.

Parameter Usage

Relevance condition(s):

Condition: Available only for grabber-controlled exposure line-scan and area-scan cameras: CamConfig must be set to PxxRG or LxxxRG

Parameter Values

OFF

Base DualBase Full FullXR

MC_InterleavedAcquisition_OFF
Description Interleaved acquisition is disabled.
Default value.

ON

Base DualBase Full FullXR

MC_InterleavedAcquisition_ON

Description
Interleaved acquisition is enabled.

ExposureTime_P1_us

Base DualBase Full FullXR

Exposure time setting for P1 program cycles

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	ADJUST	Float	Set and Get

Num ID	String Identifier	C, C++ identifier
10940 << 14	ExposureTime_P1_us	MC_ExposureTime_P1_us

Parameter Description

This parameter allows to specify the time interval between the Start of Exposure (ResetON) and the End of Exposure (ResetOFF) events of a P1 program cycle.

MultiCam calculates a default value that is equal to the largest exposure time allowed by the camera when operating at the maximum cycle rate. The maximum cycle rate is defined by LineRate_Hz for line-scan cameras and FrameRate_mHz for area-scan cameras.

The effective exposure time is reported by ExposureTime_P1_Effective_us.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

Directive: The user must set the exposure time according to the application needs within the range of values allowed by the camera. The camera exposure time range is defined by ExposeMin_us and ExposeMax_us camera parameters.

Parameter Values

Value	Description
0.16	0.16 microseconds (=160 nanoseconds) *Minimum range value.*
5000000	5,000,000 microseconds (=5 seconds) *Maximum range value.*

ExposureTime_P1_Effective_us

Base DualBase Full FullXR

Effective exposure time for P1

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Float	Get Only

Num ID	String Identifier	C, C++ identifier
11005 << 14	ExposureTime_P1_ Effective_us	MC_ExposureTime_P1_Effective_us

Parameter Description

This parameter reports the effective time interval between the Start of Exposure (ResetON) and the End of Exposure (ResetOFF) events of a P1 program cycle.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

ExposureTime_P2_us

Base DualBase Full FullXR

Exposure time setting for P2 program cycles

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	ADJUST	Float	Set and Get

Num ID	String Identifier	C, C++ identifier
10941 << 14	ExposureTime_P2_us	MC_ExposureTime_P2_us

Parameter Description

This parameter allows to specify the time interval between the Start of Exposure (ResetON) and the End of Exposure (ResetOFF) events of a P2 program cycle.

MultiCam calculates a default value that is equal to the largest exposure time allowed by the camera when operating at the maximum cycle rate. The maximum cycle rate is defined by LineRate_Hz for line-scan cameras and FrameRate_mHz for area-scan cameras.

The effective exposure time is reported by ExposureTime_P2_Effective_us.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

Directive: The user must set the exposure time according to the application needs within the range of values allowed by the camera. The camera exposure time range is defined by ExposeMin_us and ExposeMax_us camera parameters.

Parameter Values

Value	Description
0.16	0.16 microseconds (=160 nanoseconds) *Minimum range value.*
5000000	5,000,000 microseconds (=5 seconds) *Maximum range value.*

ExposureTime_P2_Effective_us

Base DualBase Full FullXR

Effective exposure time for P2

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Float	Get Only

Num ID	String Identifier	C, C++ identifier
11006 << 14	ExposureTime_P2_ Effective_us	MC_ExposureTime_P2_Effective_us

Parameter Description

This parameter reports the effective time interval between the Start of Exposure (ResetON) and the End of Exposure (ResetOFF) events of a P2 program cycle.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

ExposureDelayControl

Base DualBase Full FullXR

Control method of the exposure delay

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Float	Set and Get

Num ID	String Identifier	C, C++ identifier
11045 << 14	ExposureDelayControl	MC_ExposureDelayControl

Parameter Description

This parameter allows to select the method used by MultiCam to calculate the Exposure Delay value for P1 and P2 programs.

By default, MultiCam configures P1 and P2 with the smallest possible Exposure Delay value. This setting is satisfactory for the use cases where the exposure time is shorter than the readout time.

Optionally, keeping ExposureDelayControl set to MAN, allows to change the minimum exposure delay value of P1 and/or P2 using the ExposureDelay_MAN_P1_us and ExposureDelay_MAN_P2_us parameters.

Alternatively, you may also change ExposureDelayControl to one of the automatic control methods: SAME_START_EXPOSURE or SAME_END_EXPOSURE.

With SAME_START_EXPOSURE, the start of exposure is delayed by the same amount of time for both programs: both exposure delay values are equal.

With SAME_END_EXPOSURE the end of exposure is delayed by the same amount of time for both programs.

The effective exposure delay values, calculated by MultiCam are reported by ExposureDelay_P1_Effective_us and ExposureDelay_P2_Effective_us parameters.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

Parameter Values

MAN

Base DualBase Full FullXR

MC_ExposureDelayControl_MAN

Description
Manual control method.
The user may specify the minimum exposure delay value of P1 and/or P2 using the ExposureDelay_MAN_P1_us and ExposureDelay_MAN_P2_us parameters respectively.

Default value.

SAME_START_EXPOSURE

Base DualBase Full FullXR

MC_ExposureDelayControl_SAME_START_EXPOSURE

Description
Automatic control method 1.
The time interval from the cycle trigger to the start of exposure is identical for both programs.

SAME_START_EXPOSURE

Base DualBase Full FullXR

MC_ExposureDelayControl_SAME_START_EXPOSURE

Description
Automatic control method 2.
The time interval from the cycle trigger to the end of exposure is identical for both programs.

ExposureDelay_MAN_P1_us

Base DualBase Full FullXR

Minimum exposure delay value for P1

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Float	Set and Get

Num ID	String Identifier	C, C++ identifier
11039 << 14	ExposureDelay_MAN_P1_us	MC_ExposureDelay_MAN_P1_us

Parameter Description

When InterleavedAcquisition is set to ON, this parameter allows to specify the minimum time interval to be inserted before the Start of Exposure (ResetON) of P1 program cycles.

The effective time interval is reported by ExposureDelay_P1_Effective_us.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

Condition: ExposureDelayControl must be set to MAN.

Parameter Values

Value	Description
0	*Minimum range value.*
5000000	5,000,000 microseconds (=5 seconds) *Maximum range value.*

ExposureDelay_P1_Effective_us

Base DualBase Full FullXR

Effective exposure delay value for P1 program

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Float	Get Only

Num ID	String Identifier	C, C++ identifier
11011 << 14	ExposureDelay_P1_Effective_us	MC_ExposureDelay_P1_Effective_us

Parameter Description

This parameter reports the effective time delay inserted before the Start of Exposure (ResetON) event of P1 program cycles.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

ExposureDelay_MAN_P2_us

Base DualBase Full FullXR

Minimum exposure delay value for P2

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Float	Set and Get

Num ID	String Identifier	C, C++ identifier
11040 << 14	ExposureDelay_MAN_P2_us	MC_ExposureDelay_MAN_P2_us

Parameter Description

When InterleavedAcquisition is set to ON, this parameter allows to specify the minimum time interval to be inserted before the Start of Exposure (ResetON) of P2 program cycles.

The effective time interval is reported by ExposureDelay_P2_Effective_us.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

Condition: ExposureDelayControl must be set to MAN.

Parameter Values

Value	Description
0	*Minimum range value.*
5000000	5,000,000 microseconds (=5 seconds) *Maximum range value.*

ExposureDelay_P2_Effective_us

Base DualBase Full FullXR

Effective exposure delay value for P2 program

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Float	Get Only

Num ID	String Identifier	C, C++ identifier
11012 << 14	ExposureDelay_P2_Effective_us	MC_ExposureDelay_P2_Effective_us

Parameter Description

This parameter reports the effective time delay inserted before the Start of Exposure (ResetON) event of P2 program cycles.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

StrobeDuration_P1_us

Base DualBase Full FullXR

Strobe duration setting for P1 program cycles

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	ADJUST	Float	Set and Get

Num ID	String Identifier	C, C++ identifier
10950 << 14	StrobeDuration_P1_us	MC_StrobeDuration_P1_us

Parameter Description

This parameter allows to specify the time interval between the Start of Illumination (StrobeON) and the End of Illumination (StrobeOFF) events of a P1 program cycle.

MultiCam calculates a default value that is equal to 50% of the default exposure time.

The effective strobe duration is reported by StrobeDuration_P1_Effective_us.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

Directive: The user must set the strobe duration according to the application needs. Values larger than the exposure time are allowed.

Parameter Values

Value	Description
0.16	0.16 microseconds (= 160 nanoseconds) before *Minimum range value.*
5000000	5,000,000 microseconds (= 5 seconds) after *Maximum range value.*

StrobeDuration_P1_Effective_us

Base DualBase Full FullXR

Effective strobe duration for P1

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Float	Get Only

Num ID	String Identifier	C, C++ identifier
11007 << 14	StrobeDuration_P1_Effective_us	MC_StrobeDuration_P1_Effective_us

Parameter Description

This parameter reports the effective time interval between the Start of Illumination (StrobeON) and the End of Illumination (StrobeOFF) events of a P1 program cycle.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

StrobeDuration_P2_us

Base DualBase Full FullXR

Strobe duration setting for P2 program cycles

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	ADJUST	Float	Set and Get

Num ID	String Identifier	C, C++ identifier
10951 << 14	StrobeDuration_P2_us	MC_StrobeDuration_P2_us

Parameter Description

This parameter allows to specify the time interval between the Start of Illumination (StrobeON) and the End of Illumination (StrobeOFF) events of a P2 program cycle.

MultiCam calculates a default value that is equal to 50% of the default exposure time.

The effective strobe duration is reported by StrobeDuration_P2_Effective_us.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

Directive: The user must set the strobe duration according to the application needs. Values larger than the exposure time are allowed.

Parameter Values

Value	Description
0.16	0.16 microseconds (= 160 nanoseconds) before *Minimum range value.*
5000000	5,000,000 microseconds (= 5 seconds) after *Maximum range value.*

StrobeDuration_P2_Effective_us

Base DualBase Full FullXR

Effective strobe duration for P2

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Float	Get Only

Num ID	String Identifier	C, C++ identifier
11008 << 14	StrobeDuration_P2_Effective_us	MC_StrobeDuration_P2_Effective_us

Parameter Description

This parameter reports the effective time interval between the Start of Illumination (StrobeON) and the End of Illumination (StrobeOFF) events of a P2 program cycle.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

StrobeDelay_P1_us

Base DualBase Full FullXR

Strobe delay setting for P1

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	ADJUST	Float	Set and Get

Num ID	String Identifier	C, C++ identifier
10953 << 14	StrobeDelay_P1_us	MC_StrobeDelay_P1_us

Parameter Description

This parameter allows to specify the time interval from the Start of Exposure (ResetON) to the Start of Illumination (StrobeON) events of P1 program cycles.

The default value is 0.

The effective delay is reported by StrobeDelay_P1_Effective_us.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

Directive: The user must set the strobe delay according to the application needs. Set a positive values to retard the Start of Illumination (StrobeON) event. Set a negative values to advance the Start of Illumination (StrobeON) event.

Parameter Values

Value	Description
-10000	10,000 microseconds (= 10 milliseconds) before *Minimum range value.*
5000000	5,000,000 microseconds (= 5 seconds) after *Maximum range value.*

StrobeDelay_P1_Effective_us

Base DualBase Full FullXR

Effective strobe delay value for P1 program

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Float	Get Only

Num ID	String Identifier	C, C++ identifier
11009 << 14	StrobeDelay_P1_Effective_us	MC_StrobeDelay_P1_Effective_us

Parameter Description

This parameter reports the effective time interval from the Start of Exposure (ResetON) to the Start of Illumination (StrobeON) events of P1 program cycles.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

StrobeDelay_P2_us

Base DualBase Full FullXR

Strobe delay setting for P2

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	ADJUST	Float	Set and Get

Num ID	String Identifier	C, C++ identifier
10954 << 14	StrobeDelay_P2_us	MC_StrobeDelay_P2_us

Parameter Description

This parameter allows to specify the time interval from the Start of Exposure (ResetON) to the Start of Illumination (StrobeON) events of P2 program cycles.

The default value is 0.

The effective delay is reported by StrobeDelay_P2_Effective_us.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

Directive: The user must set the strobe delay according to the application needs. Set a positive values to retard the Start of Illumination (StrobeON) event. Set a negative values to advance the Start of Illumination (StrobeON) event.

Parameter Values

Value	Description
-10000	10,000 microseconds (= 10 milliseconds) before *Minimum range value.*
5000000	5,000,000 microseconds (= 5 seconds) after *Maximum range value.*

StrobeDelay_P2_Effective_us

Base DualBase Full FullXR

Effective strobe delay value for P2 program

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Float	Get Only

Num ID	String Identifier	C, C++ identifier
11010 << 14	StrobeDelay_P2_Effective_us	MC_StrobeDelay_P2_Effective_us

Parameter Description

This parameter reports the effective time interval from the Start of Exposure (ResetON) to the Start of Illumination (StrobeON) events of P2 program cycles.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

MinTriggerPeriod_P1_Effective_us

Base DualBase Full FullXR

Minimum time interval between cycle triggers for P1

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Float	Get Only

Num ID	String Identifier	C, C++ identifier
11035 << 14	MinTriggerPeriod_P1_Effective_us	MC_MinTriggerPeriod_P1_Effective_us

Parameter Description

This parameter reports the effective time interval between a P1 Cycle start trigger and the next cycle start trigger.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

MinTriggerPeriod_P2_Effective_us

Base DualBase Full FullXR

Minimum time interval between cycle triggers for P2

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Float	Get Only

Num ID	String Identifier	C, C++ identifier
11036 << 14	MinTriggerPeriod_P2_Effective_us	MC_MinTriggerPeriod_P2_Effective_us

Parameter Description

This parameter reports the effective time interval between a P2 Cycle start trigger and the next cycle start trigger.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

StrobeLine_P1

Base DualBase Full FullXR

Strobe output line of P1 program

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10955 << 14	StrobeLine_P1	MC_StrobeLine_P1

Parameter Description

This parameter allows to designate the output line delivering the strobe output of the P1 program.

By default, MultiCam assigns the strobe output of the P1 program to the IOUT1 output port.

NOTE
When the P1 and P2 programs are assigned to the same strobe output line,the strobe signals are logically OR-ed.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

Directive: The user may select the alternate strobe line by selecting IOUT2 or disable the strobe output of the P1 program by selecting NONE.

Parameter Values

IOUT1

Base DualBase Full FullXR

MC_StrobeLine_P1_IOUT1
Description The strobe signal of P1 program is assigned to the IOUT 1 output port.
Default value.

IOUT2

Base DualBase Full FullXR

MC_StrobeLine_P1_IOUT2

Description
The strobe signal of P1 program is assigned to the IOUT 2 output port.

NONE

Base DualBase Full FullXR

MC_StrobeLine_P1_NONE

Description
The strobe signal of P1 program is not assigned to any output port.

StrobeLine_P2

Base DualBase Full FullXR

Strobe output line of P2 program

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10956 << 14	StrobeLine_P2	MC_StrobeLine_P2

Parameter Description

This parameter allows to designate the output line delivering the strobe output of the P2 program.

By default, MultiCam assigns the strobe output of the P2 program to the IOUT2 output port.

NOTE
When the P1 and P2 programs are assigned to the same strobe output line,the strobe signals are logically OR-ed.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

Directive: The user may select the alternate strobe line by selecting IOUT2 or disable the strobe output of the P2 program by selecting NONE.

Parameter Values

IOUT1

Base DualBase Full FullXR

MC_StrobeLine_P2_IOUT1
Description The strobe signal of P2 program is assigned to the IOUT 1 output port.

IOUT2

Base DualBase Full FullXR

MC_StrobeLine_P2_IOUT2

Description
The strobe signal of P2 program is assigned to the IOUT 2 output port.

Default value.

NONE

Base DualBase Full FullXR

MC_StrobeLine_P2_NONE

Description
The strobe signal of P2 program is not assigned to any output port.

StrobeOutput_P1

Base DualBase Full FullXR

Strobe output control for P1

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11020 << 14	StrobeOutput_P1	MC_StrobeOutput_P1

Parameter Description

This parameter allows to enable or disable immediately the strobe output of the P1 program.

MultiCam enables automatically the strobe output of P1 program providing that:

- InterleavedAcquisition is set to ON,
- StrobeLine_P1 is set to IOUT1 or IOUT2.

MultiCam disables automatically the output when one of the above condition becomes false.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

Directive: If required, the user may override the parameter value at any time.

Parameter Values

ENABLE

Base DualBase Full FullXR

MC_StrobeOutput_P1_ENABLE
Description The Strobe output of P1 program is enabled.
Default value.

DISABLE

Base DualBase Full FullXR

MC_StrobeOutput_P1_DISABLE

Description
The Strobe output of P1 program is disabled.

StrobeOutput_P2

Base DualBase Full FullXR

Strobe output control for P2

Parameter Info

Class	Category	Level	Type	Access
Channel	Interleaved Acquisition	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11021 << 14	StrobeOutput_P2	MC_StrobeOutput_P2

Parameter Description

This parameter allows to enable or disable immediately the strobe output of the P2 program.

MultiCam enables automatically the strobe output of P2 program providing that:

- InterleavedAcquisition is set to ON,
- StrobeLine_P2 is set to IOUT1 or IOUT2.

MultiCam disables automatically the output when one of the above condition becomes false.

Parameter Usage

Prerequisite action(s):

Condition: InterleavedAcquisition must be set to ON.

Directive: If required, the user may override the parameter value at any time.

Parameter Values

ENABLE

Base DualBase Full FullXR

MC_StrobeOutput_P2_ENABLE
Description The Strobe output of P2 program is enabled.
Default value.

DISABLE

Base DualBase Full FullXR

MC_StrobeOutput_P2_DISABLE

Description
The Strobe output of P2 program is disabled.

4.8. Exposure Control Category

Parameters controlling the camera exposure related features associated to the channel

Expose_us

Base DualBase Full FullXR

Line or frame exposure duration, expressed in microseconds

Parameter Info

Class	Category	Level	Type	Access
Channel	Exposure Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
830 << 14	Expose_us	MC_Expose_us

Parameter Description

For area-scan cameras, the Expose_us parameter relates to the duration of the frame exposure period.

For line-scan cameras, the Expose_us parameter relates to the duration of the line exposure period.

Specifically, several controllable cameras make possible for the frame grabber to take control on the exposure period within the camera. This equally applies to the line exposure (line-scan) or frame exposure (area-scan). If an area-scan camera has this exposure control capability, and if it is configured in such a way that this capability is exercised, the camera is said to assume the grabber-controlled exposure mode.

This parameter applies only when camera operates in grabber-controlled exposure mode.

Refer to the expert-level parameters of the Camera Features Category Expose and Readout .

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 microsecond *Minimum range value.*
20000000	20,000,000 microseconds (=20 seconds) *Maximum range value.*

ExposeTrim

Base DualBase Full FullXR

Amending value for exposure duration, expressed in decibels

Parameter Info

Class	Category	Level	Type	Access
Channel	Exposure Control	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
831 << 14	ExposeTrim	MC_ExposeTrim

Parameter Description

This parameter can be used to refine the value programmed by the Expose_us parameter.

The following chart helps to understand this logarithmic control process.

Parameter Values

Base DualBase Full FullXR

Value	Description
-6	- 6 dB: The effective trimmed exposure is Expose_us x 0.5
-3	-3 dB: The effective trimmed exposure is Expose_us x 0.7
0	0 dB: The effective trimmed exposure is Expose_us *Default value.*
3	+3 dB: The effective trimmed exposure is Expose_us x 1.4
6	+6 dB: The effective trimmed exposure is Expose_us x 2.0
9	+9 dB: The effective trimmed exposure is Expose_us x 2.8
12	+12 dB: The effective trimmed exposure is Expose_us x 4.0

TrueExp_us

Base DualBase Full FullXR

Exact exposure duration, expressed in microseconds

Parameter Info

Class	Category	Level	Type	Access
Channel	Exposure Control	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
832 << 14	TrueExp_us	MC_TrueExp_us

Parameter Description

This parameter returns the effective duration of the exposure period, merging the values of Expose_us and ExposeTrim .

Some camera and/or frame grabber limitation can be such that the effective exposure duration may slightly differ from the requested exposure duration.

Setting this parameter is required when the strobe function is involved while the grabber does not positively control the exposure function. See StrobeMode .

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 microsecond *Minimum range value.*
20000000	20,000,000 microseconds (=20 seconds) *Maximum range value.*

4.9. Strobe Control Category

Parameters controlling the illumination features associated to the channel

StrobeMode

Base DualBase Full FullXR

Method for generating strobe pulse to illumination system

Parameter Info

Class	Category	Level	Type	Access
Channel	Strobe Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1197 << 14	StrobeMode	MC_StrobeMode

Parameter Description

This parameter establishes the method according to which the illumination control pulse is generated.

For area-scan cameras, this parameter relates to the illumination during the frame exposure period.

For line-scan cameras, this parameter relates to the illumination during the line exposure period.

The default value is set automatically to MAN or AUTO by the exposure controller of MultiCam.

Parameter Values

NONE

Base DualBase Full FullXR

MC_StrobeMode_NONE
Description The strobe function is disabled. No strobe line is allocated to the channel. The hardware line dedicated to issuing the strobe pulse is available for general-purpose usage.

AUTO

Base DualBase Full FullXR

MC_StrobeMode_AUTO

Description
The strobe function is enabled with an automatic timing control feature. StrobeDur and StrobePos parameters define the strobe pulse, using the exposure duration declared by the Expose_us parameter.

Base DualBase Full FullXR
Default value.

Applicability condition(s)
Condition: Grabber controlled exposure.

MAN

Base DualBase Full FullXR

MC_StrobeMode_MAN

Description
The strobe function is enabled with a manual timing control feature. StrobeDur and StrobePos parameters define the strobe pulse, using the exposure duration declared by the TrueExp_us parameter.

Base DualBase Full FullXR
Default value.

Applicability condition(s)
Condition: Camera controlled exposure.

OFF

Base DualBase Full FullXR

MC_StrobeMode_OFF

Description
The designed StrobeLine is set to the inactive level; no more strobe pulses are issued.

StrobeDur

Base DualBase Full FullXR

Duration of strobe pulse to illumination system

Parameter Info

Class	Category	Level	Type	Access
Channel	Strobe Control	ADJUST	Integer	Set and Get

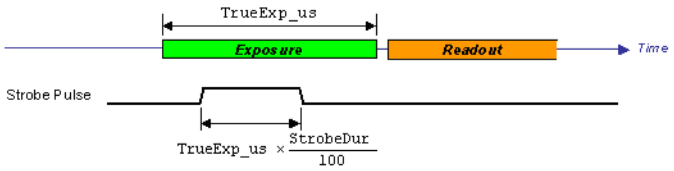
Num ID	String Identifier	C, C++ identifier
834 << 14	StrobeDur	MC_StrobeDur

Parameter Description

This parameter is expressed as a percentage of the current exposure duration as returned by TrueExp_us .

A value of 50 % means that the duration of the strobe pulse is half the duration of the exposure period.

- For area-scan cameras, the StrobeDur parameter relates to the illumination during the frame exposure period.
- For line-scan cameras, the StrobeDur parameter relates to the illumination during the line exposure period.



Strobe duration formula

StrobePos

Base DualBase Full FullXR

Position of strobe pulse to illumination system

Parameter Info

Class	Category	Level	Type	Access
Channel	Strobe Control	ADJUST	Integer	Set and Get

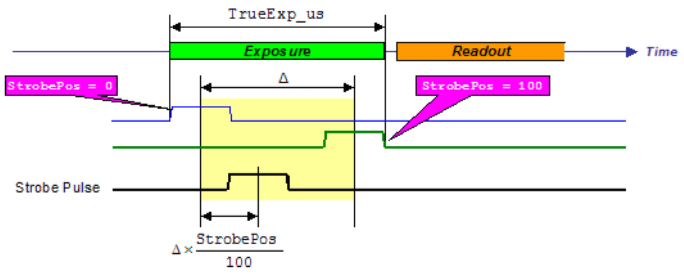
Num ID	String Identifier	C, C++ identifier
835 << 14	StrobePos	MC_StrobePos

Parameter Description

This parameter is expressed as a percentage of the allowed position range of the strobe pulse for its current duration.

A value of 0 % establishes the earliest position. A value of 100 % establishes the latest position.

A value of 50 % means that the strobe pulse is located in the middle of the exposure period.



Strobe Position vs. StrobePos value

- For area-scan cameras, the StrobePos parameter relates to the illumination during the frame exposure period.
- For line-scan cameras, the StrobePos parameter relates to the illumination during the line exposure period.

$$20 \times \log \frac{T_{Trimmed}}{T_{Exposure}}$$

Strobe position formula

The StrobePos refers to the middle of the strobe pulse.

StrobeCtl

Base DualBase Full FullXR

Electrical style of designated strobe pulse to illumination system

Parameter Info

Class	Category	Level	Type	Access
Channel	Strobe Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
836 << 14	StrobeCtl	MC_StrobeCtl

Parameter Description

This parameter declares the attributes of the strobe line designated by StrobeLine sent by the channel and aimed at generating an illumination pulse.

Parameter Values

OPTO

Base DualBase Full FullXR

MC_StrobeCtl_OPTO
Description The strobe line is issued on an opto-isolated pair of pins. The + pin is the collector and the - pin is the emitter of an uncommitted photo-transistor driven by LED-emitted light.

PreStrobe_us

Base DualBase Full FullXR

Time delay, expressed in microseconds, before the pulse defined by StrobePos

Parameter Info

Class	Category	Level	Type	Access
Channel	Strobe Control	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
2190 << 14	PreStrobe_us	MC_PreStrobe_us

Parameter Description

This parameter declares the delay of the beginning of the strobe pulse before the normal beginning defined by StrobePos . If long enough, it creates a "pre-exposure" phase before actual "start of exposure phase" (SAP).

- For area-scan cameras, this parameter relates to the illumination during the frame exposure period.
- For line-scan cameras, this parameter is irrelevant.

Parameter Values

Base DualBase Full FullXR

Value	Description
0	*Minimum range value.*
10000	10,000 microseconds (=10 milliseconds) *Maximum range value.*

4.10. Encoder Control Category

Parameters controlling the motion encoder rate conversion device embedded in the line-scan capable frame grabbers

LineCaptureMode

Base DualBase Full FullXR

Fundamental line capturing mode

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
3001 << 14	LineCaptureMode	MC_LineCaptureMode

Parameter Description

In line-scan system, this parameter declares the **fundamental line capturing mode**.

Parameter Usage

Relevance condition(s):

Condition: Any line-scan acquisition mode.

Parameter Values

ALL

Base DualBase Full FullXR

MC_LineCaptureMode_ALL
Description **Take-All-Lines** capture mode. Each delivered camera line results into a line acquisition. This is the traditional operating mode. If the down-web motion speed is varying, the line-scanning process of the camera would be rate-controlled accordingly.
Default value.

PICK

Base DualBase Full FullXR

MC_LineCaptureMode_PICK
Description **Pick-A-Line** line capture mode. The line-scanning process of the camera is running at a constant rate. Each pulse occurring at the down-web line rate determines the acquisition of the next line delivered by the camera.

TAG

Base DualBase Full FullXR

MC_LineCaptureMode_TAG

Description

Tag-A-Line capture mode. The line-scanning process of the camera is running at a constant rate determined by Period_us.

The down-web line rate is determined by the pulse rate of A/B signals delivered by an external encoder and processed by the quadrature decoder and the rate divider.

The frame grabber captures all lines delivered by the camera after having replaced the first pixel data by a **tag** indicating that the line was preceded or not by an hardware event on the divider output.

LineRateMode

Base DualBase Full FullXR

Line rate generation method

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1328 << 14	LineRateMode	MC_LineRateMode

Parameter Description

In line-scan system, this parameter declares the device responsible for line rate generation.

For more information, refer to "Line Rate Modes" on page 510.

When LineRateMode is set to PERIOD, the downweb line rate is controlled by the Period_us parameter.

When LineRateMode is set to EXPOSE, the downweb line rate is controlled by the Expose_us parameter.

When LineRateMode is set to PULSE or CONVERT, the downweb line rate is directed by a pulse signal applied to line trigger hardware line selected by the LineTrigLine parameter.

The applicable line rate modes are depending on the selected LineCaptureMode and on the camera line-scanning mode. The camera line-scanning mode is determined by the Expose and Readout parameters ("Camera Features Category " on page 108). Two classes of camera line-scanning mode are considered in this case:

- Free-running cameras
- Controlled line rate cameras

Parameter Values

CONVERT

Base DualBase Full FullXR

MC_LineRateMode_CONVERT
Description **Rate Converter**. The downweb line rate is derived from a train of trigger pulses processed by a rate converter belonging to the grabber.

PULSE

Base DualBase Full FullXR

MC_LineRateMode_PULSE

Description
Trigger Pulse. The downweb line rate is directly derived from trigger pulses applied to the grabber.

PERIOD

Base DualBase Full FullXR

MC_LineRateMode_PERIOD

Description
Periodic. The downweb line rate is internally generated by a periodic generator.

CAMERA

Base DualBase Full FullXR

MC_LineRateMode_CAMERA

Description
Camera. The downweb line rate is originated from the camera.

EXPOSE

Base DualBase Full FullXR

MC_LineRateMode_EXPOSE

Description
Exposure Time. The downweb line rate is identical to the camera line rate, and established by the exposure time settings.

SLAVE

Base DualBase Full FullXR

MC_LineRateMode_SLAVE

Description
Slave. The downweb line rate is originated from the master device. LineRateMode is automatically set to this value set when SynchronizedAcquisition = SLAVE or LOCAL_SLAVE.

Period_us

Base DualBase Full FullXR

Programmable line-scan period, expressed in microseconds

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
1329 << 14	Period_us	MC_Period_us

Parameter Description

This parameter allows for programming the periodic generator issuing the downweb line rate in line-scan systems.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 microsecond *Minimum range value.*
1000000	1,000,000 microseconds (=1 second) *Maximum range value.*

PeriodTrim

Base DualBase Full FullXR

Amending value for line-scan period duration

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1330 << 14	PeriodTrim	MC_PeriodTrim

Parameter Description

This parameter can be used to refine the value programmed by the Period_us parameter.

Parameter Values

Base DualBase Full FullXR

Value	Description
-6	-6 dB: The effective trimmed period is Period_us x 0.5
-3	-3 dB: The effective trimmed period is Period_us x 0.7
0	0 dB: The effective trimmed period is Period_us *Default value.*
3	+3 dB: The effective trimmed period is Period_us x 1.4
6	+6 dB: The effective trimmed period is Period_us x 2.0
9	+9 dB: The effective trimmed period is Period_us x 2.8
12	+12 dB: The effective trimmed period is Period_us x 4.0

LinePitch

Base DualBase Full FullXR

Line pitch for rate converter programming

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
595 << 14	LinePitch	MC_LinePitch

Parameter Description

This parameter applies when the motion encoder is in use with rate conversion.

The parameter declares in an arbitrary length unit the distance between two successively scanned lines on the observed moving web.

Along with EncoderPitch , it allows for programming the rate converter issuing the line rate in line-scan systems. The EncoderPitch parameter should be expressed in the same length unit.

The programmed rate conversion ratio is: *RateConversionRatio = EncoderPitch/LinePitch*.

The resulting downweb line rate is: *DownwebLineRate = EncoderRate x RateConversioRatio*

The encoder rate at a given time is the frequency of the pulses delivered by the motion encoder while the observed web is moving.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 length unit *Minimum range value.*
10000	10,000 length units *Maximum range value.*

EncoderPitch

Base DualBase Full FullXR

Encoder pitch for rate converter programming

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
592 << 14	EncoderPitch	MC_EncoderPitch

Parameter Description

This parameter applies when the motion encoder is in use with rate conversion.

The parameter declares in an arbitrary length unit the distance traveled between two successive pulses issued by the motion encoder.

Along with LinePitch, it allows for programming the rate converter issuing the line rate in line-scan systems. The LinePitch parameter should be expressed in the same length unit.

The programmed rate conversion ratio is: *RateConversionRatio = EncoderPitch/LinePitch*.

The resulting downweb line rate is: *DownwebLineRate = EncoderRate x RateConversioRatio*

The encoder rate at a given time is the frequency of the pulses delivered by the motion encoder while the observed web is moving.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 length unit *Minimum range value.*
10000	10,000 length units *Maximum range value.*

LineTrigCtl

Base DualBase Full FullXR

Electrical style of designated line trigger hardware line from outside system

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1331 << 14	LineTrigCtl	MC_LineTrigCtl

Parameter Description

This parameter applies to the hardware line designated by LineTrigLine when LineRateMode is set to PULSE or CONVERT.

Along with LineTrigEdge and LineTrigFilter, it declares the grabber attributes of the terminal sensing the line trigger pulse. The line trigger pulse is processed by the grabber and transferred to the camera as the line reset pulse.

The LineTrigCtl parameter determines the electrical style of the line trigger pulse usually issued by a motion encoder.

Parameter Values

DIFF

Base DualBase Full FullXR

MC_LineTrigCtl_DIFF
Description Differential high-speed input compatible with EIA/TIA-422 signaling.

DIFF_PAIRED

Base DualBase Full FullXR

MC_LineTrigCtl_DIFF_PAIRED
Description Dual differential high-speed input compatible with EIA/TIA-422 signaling.
Default value.

ISO

Base DualBase Full FullXR

MC_LineTrigCtl_ISO

Description
Isolated current loop input compatible with TTL, +12V, +24V signaling.

ISO_PAIRED

Base DualBase Full FullXR

MC_LineTrigCtl_ISO_PAIRED

Description
Dual isolated current loop input compatible with TTL, +12V, +24V signaling.

LineTrigEdge

Base DualBase Full FullXR

Significant edge of designated line trigger hardware line from outside system

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1332 << 14	LineTrigEdge	MC_LineTrigEdge

Parameter Description

This parameter applies to the hardware line designated by LineTrigLine when LineRateMode is set to PULSE or CONVERT.

Along with LineTrigCtl and LineTrigFilter, it declares the grabber attributes of the terminal sensing the line trigger pulse. The line trigger pulse is processed by the grabber and transferred to the camera as the line reset pulse.

The LineTrigEdge parameter determines the significant edge of the line trigger pulse usually issued by a motion encoder.

Parameter Values

GOHIGH

Base DualBase Full FullXR

MC_LineTrigEdge_GOHIGH

Base DualBase Full FullXR

Description
Value equivalent to RISING_A.

GOLOW

Base DualBase Full FullXR

MC_LineTrigEdge_GOLOW

Base DualBase Full FullXR

Description
Value equivalent to FALLING_A.

RISING_A

Base DualBase Full FullXR

MC_LineTrigEdge_RISING_A

Description
An output pulse is generated for every rising edge of the A signal. The falling edge on the A signal and both edges on the B-signal are ignored.

FALLING_A

Base DualBase Full FullXR

MC_LineTrigEdge_FALLING_A

Description
An output pulse is generated for every falling edge of the A signal. The rising edge on the A signal and both edges on the B-signal are ignored.

ALL_A

Base DualBase Full FullXR

MC_LineTrigEdge_ALL_A

Description
An output pulse is generated for every rising and falling edges of the A signal. The B-signal is ignored.

Applicability condition(s)
Condition: LineTrigCtl is set to DIFF or ISO.

ALL_A_B

Base DualBase Full FullXR

MC_LineTrigEdge_ALL_A_B

Description
An output pulse is generated for every rising and falling edges of the A and B signals.

Applicability condition(s)
Condition: LineTrigCtl is set to DIFF_PAIRED or ISO_PAIRED.

LineTrigFilter

Base DualBase Full FullXR

Noise removal on designated line trigger hardware line from outside system

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1333 << 14	LineTrigFilter	MC_LineTrigFilter

Parameter Description

This parameter applies to the hardware line designated by LineTrigLine when LineRateMode is set to PULSE or CONVERT.

Along with LineTrigCtl and LineTrigEdge, it declares the grabber attributes of the terminal sensing the line trigger pulse. The line trigger pulse is processed by the grabber and transferred to the camera as the line reset pulse.

The LineTrigFilter parameter reduces the noise sensitivity over the line trigger pulse usually issued by a motion encoder.

The time constant of the filter is the amount of time the line should be detected at the same logic state before a logic transition be considered.

When LineTrigLine is an insulated I/O, LineTrigFilter is forced to STRONG.

Parameter Values

OFF

Base DualBase Full FullXR

MC_LineTrigFilter_OFF

Base DualBase Full FullXR

Description
The filter time constant is approximately 40 ns.

MEDIUM

Base DualBase Full FullXR

MC_LineTrigFilter_MEDIUM

Base DualBase Full FullXR

Description
The filter time constant is approximately 500 ns.

Default value.

STRONG

Base DualBase Full FullXR

MC_LineTrigFilter_STRONG

Base DualBase Full FullXR

Description
The filter time constant is approximately 5 µs.

Filter_40ns

Base DualBase Full FullXR

MC_LineTrigFilter_Filter_40ns

Description
The filter time constant is approximately 40 ns.

Filter_100ns

Base DualBase Full FullXR

MC_LineTrigFilter_Filter_100ns

Description
The filter time constant is approximately 100 ns.

Filter_200ns

Base DualBase Full FullXR

MC_LineTrigFilter_Filter_200ns

Description
The filter time constant is approximately 200 ns.

Filter_500ns

Base DualBase Full FullXR

MC_LineTrigFilter_Filter_500ns

Description
The filter time constant is approximately 500 ns.

Filter_1us

Base DualBase Full FullXR

MC_LineTrigFilter_Filter_1us

Description
The filter time constant is approximately 1 us.

Filter_5us

Base DualBase Full FullXR

MC_LineTrigFilter_Filter_5us

Description
The filter time constant is approximately 5 us.

Filter_10us

Base DualBase Full FullXR

MC_LineTrigFilter_Filter_10us

Description
The filter time constant is approximately 10 us.

BackwardMotionCancellationMode

Base DualBase Full FullXR

Operational mode of the Backward Motion Cancellation circuit

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10352 << 14	BackwardMotionCancellationMode	MC_BackwardMotionCancellationMode

Parameter Description

The backward cancellation circuit stops sending line trigger pulses as soon as a backward motion is detected. If such an event occurs, the acquisition is stopped.

When the backward cancellation control is configured in the FILTERED mode (F-mode), the line acquisition resumes as soon as the motion is again in the forward direction. Therefore, the cancellation circuit filters out all the pulses corresponding to the backward direction.

When the backward cancellation control is configured in the COMPENSATE mode (C-mode), the line acquisition resumes when the motion is again in the forward direction at the place it was interrupted. Therefore, the cancellation circuit filters out not only the pulses corresponding to the backward direction, but a number of forward pulses equal to the number of skipped backward pulses.

In C-Mode, the cancellation circuit uses a "backward pulse counter" that:

- Increments by 1 every clock in the backward direction
- Decrements by 1 every clock in the forward direction until it reaches 0
- Resets at the beginning of each MultiCam acquisition sequence, more precisely, at the first trigger event of the sequence. This trigger is considered as the reference for the position along the web for the whole acquisition sequence.

In C-Mode, all pulses occurring when the counter value is different of zero are blocked.

The counter has a 16-bit span; backward displacement up to 65535 pulses can be compensated.

Parameter Usage

Relevance condition(s):

Condition: The line trigger originates from a quadrature motion encoder.

Condition: The rate converter circuit is unused.

Parameter Values

OFF

Base DualBase Full FullXR

MC_BackwardMotionCancellationMode_OFF

Description
The backward motion cancellation circuit is disabled.

Default value.

FILTERED

Base DualBase Full FullXR

MC_BackwardMotionCancellationMode_FILTERED

Description
The backward motion cancellation circuit is enabled and configured for the filter mode.

Applicability condition(s)
Condition: LineRateMode is set to PULSE
Condition: LineTrigCtl is set to DIFF_PAIRED or ISO_PAIRED

COMPENSATE

Base DualBase Full FullXR

MC_BackwardMotionCancellationMode_COMPENSATE

Description
The backward motion cancellation circuit is enabled and configured for the compensation mode.

Applicability condition(s)
Condition: LineRateMode is set to PULSE
Condition: LineTrigCtl is set to DIFF_PAIRED or ISO_PAIRED

ForwardDirection

Base DualBase Full FullXR

Motion direction, determined by the phase relationship of the A and B signals

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10350 << 14	ForwardDirection	MC_ForwardDirection

Parameter Description

The motion direction is determined by the phase relationship of the A and B signals.

By construction, the dual-output phase quadrature incremental motion encoder maintains a phase relationship of about 90 degrees between the two signals. For motion in one direction, the A signal leads the B signal by about 90 degrees; for a motion in the other direction, the B signal leads the A signal by about 90 degrees.

The direction selector provides the capability to define which one of the phase relationships is considered as the forward direction for the application.

Parameter Values

A_LEADS_B

Base DualBase Full FullXR

MC_ForwardDirection_A_LEADS_B
Description The A signal leads the B signal by about 90 degrees.
Default value.

B_LEADS_A

Base DualBase Full FullXR

MC_ForwardDirection_B_LEADS_A
Description The B signal leads the A signal by about 90 degrees.

RateDivisionFactor

Base DualBase Full FullXR

Division factor of the line trigger rate divider

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
10553 << 14	RateDivisionFactor	MC_RateDivisionFactor

Parameter Description

The rate divider circuit generates a line trigger signal at a frequency that is an integer fraction 1/N of the frequency of the pulses delivered by the quadrature decoder circuit.

For N consecutive incoming pulses issued by the quadrature decoder circuit, the 1/N rate divider:

- Generates one output pulse (one line trigger)
- Skips N-1 input pulse

The rate divider is initialized at the beginning of every MultiCam acquisition sequence. The first output pulse is produced from the first clock input pulse occurring after the sequence trigger event.

Notice that:

- The output frequency is lower than (N > 1) or equal to (N = 1) the input frequency. It cannot be higher.
- The output pulse is generated with a small fixed delay after a non-skipped input pulse. The line trigger pulses are phase-locked to the quadrature decoder output.
- The rate divider settings may not be modified while acquisition is in progress.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	No division *Minimum range value. Default value.*
512	Divide by 512 *Maximum range value.*

LineTrigLine

Base DualBase Full FullXR

Designation of line trigger hardware line from outside system

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1334 << 14	LineTrigLine	MC_LineTrigLine

Parameter Description

This parameter designates the terminal sensing the line trigger pulse. The line trigger pulse is processed by the grabber and transferred to the camera as the line reset pulse. Usually, this line trigger signal is generated by a motion encoder.

The LineTrigLine parameter designates where the line trigger pulse usually issued by a motion encoder should be applied.

Parameter Values

NOM

Base DualBase Full FullXR

MC_LineTrigLine_NOM
Base DualBase Full FullXR *Description* t designates the DIN1 line when LineTrigCtl =DIFF, the DIN1 and DIN2 pair of differential lines when LineTrigCtl =DIFF_PAIRED, the IIN1 line when LineTrigCtl =ISO, the IIN1 and IIN2 pair of differential lines when LineTrigCtl =ISO_PAIRED.
Default value.
Applicability condition(s) Base DualBase Full FullXR *Condition:* LineTrigCtl is set to DIFF, DIFF_PAIRED, ISO, or ISO_PAIRED.

DIN1

Base DualBase Full FullXR

MC_LineTrigLine_DIN1

Description

Applicability condition(s)
Condition: LineTrigCtl is set to DIFF.

DIN2

Base DualBase Full FullXR

MC_LineTrigLine_DIN2

Description

Applicability condition(s)
Condition: LineTrigCtl is set to DIFF.

DIN1_DIN2

Base DualBase Full FullXR

MC_LineTrigLine_DIN1_DIN2

Description
The pair of differential input lines DIN1 and DIN2.

Applicability condition(s)
Condition: LineTrigCtl is set to DIFF_PAIRED.

IIN1

Base DualBase Full FullXR

MC_LineTrigLine_IIN1

Description

Applicability condition(s)
Condition: LineTrigCtl is set to ISO.

IIN2

Base DualBase Full FullXR

MC_LineTrigLine_IIN2

Description

Applicability condition(s)
Condition: LineTrigCtl is set to ISO.

IIN3

Base DualBase Full FullXR

MC_LineTrigLine_IIN3

Description

Applicability condition(s)
Condition: LineTrigCtl is set to ISO.

IIN4

Base DualBase Full FullXR

MC_LineTrigLine_IIN4

Description

Applicability condition(s)
Condition: LineTrigCtl is set to ISO.

IIN1_IIN2

Base DualBase Full FullXR

MC_LineTrigLine_IIN1_IIN2

Description
The pair of differential input lines IIN1 and IIN2.

Applicability condition(s)
Condition: LineTrigCtl is set to ISO_PAIRED.

IIN3_IIN4

Base DualBase Full FullXR

MC_LineTrigLine_IIN3_IIN4

Description
The pair of differential input lines IIN3 and IIN4.

Applicability condition(s)
Condition: LineTrigCtl is set to ISO_PAIRED.

EncoderTickCount

Base DualBase Full FullXR

Encoder tick counter

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
11189 << 14	EncoderTickCount	MC_EncoderTickCount

Parameter Description

This parameter applies when a motion encoder is used.

The encoder tick counter is a 32-bit binary up/down counter that counts encoder ticks delivered by the quadrature decoder.

When the quadrature decoder is configured for 2 signals, namely when LineTrigCtl is set to DIFF_PAIRED or ISO_PAIRED, the counter is incremented or decremented according to the detected motion direction. The forward direction is defined by ForwardDirection.

When the quadrature decoder is configured for 1 signal, namely when LineTrigCtl is set to DIFF or ISO, the counter is incremented only.

The number of ticks per encoder signal(s) cycle can be 1, 2 or 4 according to the value of LineTrigEdge.

The counter cannot be disabled. Reading EncoderTickCount reports the current counter value. Setting EncoderTickCount to 0 resets the counter.

The counter is automatically reset at channel activation.

Parameter Values

Value	Description
MC_MIN_INT32	2,147,483,648 encoder ticks below 0 *Minimum range value.*
MC_MAX_INT32	2,147,483,647 encoder ticks above 0 *Maximum range value.*

BMCRestart

Base DualBase Full FullXR

Restart condition of the backward motion cancellation circuit

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11187 << 14	BMCRestart	MC_BMCRestart

Parameter Description

This parameter defines when the backward motion cancellation circuit restarts.

On a restart, the backward motion cancellation circuit forgets any motion history.

Parameter Values

NEVER

Base DualBase Full FullXR

MC_BMCRestart_NEVER
Description The backward motion cancellation circuit never restarts.
Default value.

START_OF_SCAN

Base DualBase Full FullXR

MC_BMCRestart_START_OF_SCAN
Description The backward motion cancellation circuit restarts at each start-of-scan i.e. at each page in PAGE acquisition mode and at each sequence in WEB and LONGPAGE acquisition modes.

RateDividerRestart

Base DualBase Full FullXR

Restart condition of the rate divider

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11046 << 14	RateDividerRestart	MC_RateDividerRestart

Parameter Description

This parameter defines when the rate divider circuit restarts.

Parameter Values

NEVER

Base DualBase Full FullXR

MC_RateDividerRestart_NEVER

Description
The rate divider is never reinitialized.

NOTE
Default value in MultiCam 6.9.6 and older.

START_OF_SCAN

Base DualBase Full FullXR

MC_RateDividerRestart_START_OF_SCAN

Description
The rate divider is reinitialized at each start-of-scan i.e. at each page in PAGE acquisition mode and at each sequence in WEB and LONGPAGE acquisition modes.

Default value.

MaxSpeed

Base DualBase Full FullXR

Maximum operating speed of the line-scan system, expressed in Hertz

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
1336 << 14	MaxSpeed	MC_MaxSpeed

Parameter Description

This parameter applies when the motion encoder is in use with rate conversion.

The MaxSpeed parameter declares the maximum speed at which the line-scan system will be used, that is the upper limit of the rate converter operating range.

The downweb line rate has been chosen as a measurement of the system operating speed.

After programming the rate converter, this parameter is automatically set to the highest possible downweb line rate.

If desired, the parameter can be set to a lower value to reflect the actual maximum operating speed.

When LineCaptureMode = ADR, the highest possible downweb line rate can exceed the highest possible camera rate.

When LineCaptureMode = PICK or ALL, the highest possible downweb line rate is determined by the highest possible camera line rate, as indicated by LineRate_Hz .

The lower limit of the rate converter operating range is returned by the MinSpeed parameter.

The effective upper limit of the rate converter operating range is returned by the MaxSpeedEffective parameter.

Parameter Values

Base DualBase Full FullXR

Value	Description
10	10 Hz *Minimum range value.*
100000	100,000 Hz (=100 kHz) *Maximum range value.*

MaxSpeedEffective

Base DualBase Full FullXR

Effective upper limit of the rate converter output frequency, expressed in Hertz.

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
1421 << 14	MaxSpeedEffective	MC_MaxSpeedEffective

MinSpeed

Base DualBase Full FullXR

Minimum operating speed of the line-scan system, expressed in Hertz

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
1337 << 14	MinSpeed	MC_MinSpeed

Parameter Description

This parameter applies when the motion encoder is in use with rate conversion.

This parameter returns the lower limit of the line-scan system operating range.

The downweb line rate has been chosen as a measurement of the system operating speed.

The MinSpeed parameter declares the minimum downweb line rate the converter is able to support. The OnMinSpeed parameter declares the behavior of the rate converter when it reaches the bottom speed limit.

The maximum speed at which the line-scan camera will be used has to be previously declared with MaxSpeed , that sets the upper limit of the rate converter operating range.

OnMinSpeed

Base DualBase Full FullXR

Rate converter behavior below minimum speed limit

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
3374 << 14	OnMinSpeed	MC_OnMinSpeed

Parameter Description

This parameter declares the behavior of the rate converter when it reaches the bottom speed limit of the incoming line trigger rate.

Parameter Values

IDLING

Base DualBase Full FullXR

MC_OnMinSpeed_IDLING
Description The rate converter outputs trigger pulse at a frequency specified by MinSpeed when the incoming line trigger rate is below the input range.
Default value.

MUTING

Base DualBase Full FullXR

MC_OnMinSpeed_MUTING
Description The rate converter does not output trigger pulse when the incoming line trigger rate is below the input range.

CrossPitch

Base DualBase Full FullXR

Distance between two locations focusing on adjacent pixels on the CCD sensor

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
3004 << 14	CrossPitch	MC_CrossPitch

Parameter Description

This parameter states the crossweb resolution.

The LinePitch parameter expresses the downweb resolution in arbitrary length units. The "cross pitch" is defined as the crossweb resolution in the same units. This is the distance between two locations focusing on adjacent pixels on the CCD sensor.

The ratio of LinePitch and CrossPitch is nothing else than the pixel aspect ratio of the rendered image.

Each time the LinePitch parameter is set, the CrossPitch parameter will be set to the same value. This will encourage the 1-to-1 aspect ratio.

If the user expects non-square pixels, he will adjust the cross pitch after LinePitch has been set.

SynchronizedPeriodicGenerator

Base DualBase Full FullXR

Periodic Generator timer synchronization control

Parameter Info

Class	Category	Level	Type	Access
Channel	Encoder Control	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11224 << 14	SynchronizedPeriodicGenerator	MC_SynchronizedPeriodicGenerator

Parameter Description

This parameter controls the synchronization source of the Periodic Generator timer.

Parameter Usage

Prerequisite action(s):

Condition: LineRateMode = Period

Parameter Values

OFF

Base DualBase Full FullXR

MC_SynchronizedPeriodicGenerator_OFF
Description The Periodic Generator timer delivers pulses continuously while the channel is in the ACTIVE state.
Default value.

PAGETRIGGER

Base DualBase Full FullXR

MC_SynchronizedPeriodicGenerator_PAGETRIGGER
Description The Periodic Generator timer is synchronized on page triggers. It starts delivering pulses when receiving a page trigger and stops delivering pulses after acquiring the last line of the corresponding page.

4.11. Pipeline Control Category

Parameters controlling the line-scan pipeline controller

Pipeline_Control

Base DualBase Full FullXR

Master control switch of the pipeline controller feature

Parameter Info

Class	Category	Level	Type	Access
Channel	Pipeline Control	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11067 << 14	Pipeline_Control	MC_Pipeline_Control

Parameter Description

This parameter allows to enable or disable the pipeline controller feature.

When enabled, for each object passing in front of the detector, the pipeline controller:

- asserts a start-of-scan trigger after Pipeline_StartOfScan_Position encoder ticks,
- only when the application requests an ACTIVE output action, asserts a pulse on the selected pipeline output line after Pipeline_Output_Position encoder ticks.

Parameter Usage

Relevance condition(s):

Condition: Available only when the board class parameter BoardTopology is set to MONO_OPT1, DUO_OPT1 or MONO_DECA_OPT1.

Parameter Values

ENABLE

Base DualBase Full FullXR

MC_Pipeline_Control_ENABLE
Description The pipeline controller is enabled.

DISABLE

Base DualBase Full FullXR

MC_Pipeline_Control_DISABLE
Description The pipeline controller is disabled.
Default value.

Pipeline_StartOfScan_Position

Base DualBase Full FullXR

Start-of-scan position offset

Parameter Info

Class	Category	Level	Type	Access
Channel	Pipeline Control	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
11078 << 14	Pipeline_StartOfScan_Position	MC_Pipeline_StartOfScan_Position

Parameter Description

Position the FOV (Field of View) relative to the trigger sensor position, expressed as an integer number of encoder ticks.

Parameter Usage

Relevance condition(s):

Condition: Pipeline_Control = ENABLE.

Parameter Values

Value	Description
0	*Minimum range value.*
1073741823	1,073,741,823 encoder ticks *Maximum range value.*

Pipeline_Output_Position

Base DualBase Full FullXR

Output pulse position offset

Parameter Info

Class	Category	Level	Type	Access
Channel	Pipeline Control	ADJUST	Integer collection	Set and Get

Num ID	String Identifier	C, C++ identifier
11079 << 14	Pipeline_Output_Position	MC_Pipeline_Output_Position

Parameter Description

Position the output action pulse relative to the trigger sensor position, expressed as an integer number of encoder ticks.

Parameter Usage

Relevance condition(s):

Condition: Pipeline_Control = ENABLE.

Parameter Values

Value	Description
0	*Minimum range value.*
1073741823	1,073,741,823 encoder ticks *Maximum range value.*

Pipeline_Output_PulseWidth

Base DualBase Full FullXR

Pipeline controller output pulse width

Parameter Info

Class	Category	Level	Type	Access
Channel	Pipeline Control	ADJUST	Enumerated collection	Set and Get

Num ID	String Identifier	C, C++ identifier
11070 << 14	Pipeline_Output_PulseWidth	MC_Pipeline_Output_PulseWidth

Parameter Description

Time duration of the pipeline output action pulse.

Parameter Values

100us

Base DualBase Full FullXR

MC_Pipeline_Output_PulseWidth_100us
Description 100 microseconds.

200us

Base DualBase Full FullXR

MC_Pipeline_Output_PulseWidth_200us
Description 200 microseconds.

500us

Base DualBase Full FullXR

MC_Pipeline_Output_PulseWidth_500us
Description 500 microseconds.

1ms

Base DualBase Full FullXR

MC_Pipeline_Output_PulseWidth_1ms

Description
1 millisecond.

2ms

Base DualBase Full FullXR

MC_Pipeline_Output_PulseWidth_2ms

Description
2 milliseconds.

5ms

Base DualBase Full FullXR

MC_Pipeline_Output_PulseWidth_5ms

Description
5 milliseconds.

Default value.

EncTicks

Base DualBase Full FullXR

MC_Pipeline_Output_PulseWidth_EncTicks

Description
The output pulse width is determined by the value of the Pipeline_Output_PulseWidth_EncTicks parameter

Pipeline_Output_PulseWidth_EncTicks

Base DualBase Full FullXR

Pipeline controller output pulse width, expressed in encoder ticks

Parameter Info

Class	Category	Level	Type	Access
Channel	Pipeline Control	ADJUST	Integer collection	Set and Get

Num ID	String Identifier	C, C++ identifier
11220 << 14	Pipeline_Output_PulseWidth_EncTicks	MC_Pipeline_Output_PulseWidth_EncTicks

Parameter Description

Time duration of the pipeline output action pulse, expressed in encoder ticks.

Parameter Usage

Prerequisite action(s):

Condition: Pipeline_Output_PulseWidth = EncTicks

Pipeline_Output_Action

Base DualBase Full FullXR

Action to execute on the selected pipelined output

Parameter Info

Class	Category	Level	Type	Access
Channel	Pipeline Control	ADJUST	Enumerated collection	Set Only

Num ID	String Identifier	C, C++ identifier
11071 << 14	Pipeline_Output_Action	MC_Pipeline_Output_Action

Parameter Description

Defines the action that the pipeline controller must execute when the object has travelled Pipeline_Output_Position encoder ticks since the detector position.

Parameter Usage

Directive: For each acquired image scan when Pipeline_Control = ENABLE, the application has to define the action to perform on the pipeline output line according to the result of the image analysis.

Parameter Values

ACTIVE

Base DualBase Full FullXR

MC_Pipeline_Output_Action_ACTIVE

Description
Assert a pulse when the object has travelled Pipeline_Output_Position encoder ticks since the detector position.

INACTIVE

Base DualBase Full FullXR

MC_Pipeline_Output_Action_INACTIVE

Description
Don't assert a pulse when the object has travelled Pipeline_Output_Position encoder ticks since the detector position.

NONE

Base DualBase Full FullXR

MC_Pipeline_Output_Action_NONE
Description
Default value.

Pipeline_Output_Line

Base DualBase Full FullXR

GPIO output line used for pipeline control

Parameter Info

Class	Category	Level	Type	Access
Channel	Pipeline Control	ADJUST	Enumerated collection	Set and Get

Num ID	String Identifier	C, C++ identifier
11072 << 14	Pipeline_Output_Line	MC_Pipeline_Output_Line

Parameter Description

Selects the GPIO output line used by the pipeline controller to execute actions.

This is automatically connected to IOUT2 when the pipeline controller is enabled.

Parameter Usage

Directive: MultiCam automatically enforces the IOUT2 value when the application sets Pipeline_Control to ENABLE.

Parameter Values

IOUT2

Base DualBase Full FullXR

MC_Pipeline_Output_Line_IOUT2
Description Isolated Output 2 of the MultiCam Channel.

NONE

Base DualBase Full FullXR

MC_Pipeline_Output_Line_NONE
Description No GPIO line used for pipeline control output action.
Default value.

Pipeline_Fifo_Overflow

Base DualBase Full FullXR

Count of FIFO overflow errors

Parameter Info

Class	Category	Level	Type	Access
Channel	Pipeline Control	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
11073 << 14	Pipeline_Fifo_Overflow	MC_Pipeline_Fifo_Overflow

Parameter Description

Reports the number of FIFO overflow errors encountered by the pipeline controller.

An overflow occurs when too many triggers have been received or too many actions have been posted by the application software and the corresponding objects have not yet reached the output.

NOTE
The pipeline controller manages up to 32 objects in the machine pipeline.

NOTE
This counter is never reset during the lifetime of the acquisition channel.

Parameter Usage

Directive: To recover from this error, it is required to terminate the current acquisition sequence and restart a new one.

Parameter Values

Value	Description
0	No occurence *Minimum range value. Default value.*

Pipeline_Fifo_Underflow

Base DualBase Full FullXR

Count of FIFO underflow errors

Parameter Info

Class	Category	Level	Type	Access
Channel	Pipeline Control	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
11074 << 14	Pipeline_Fifo_Underflow	MC_Pipeline_Fifo_Underflow

Parameter Description

Reports the number of FIFO underflow errors encountered by the pipeline controller.

An underflow occurs when an object arrives at the end pipeline and the application software has not yet posted the output action.

NOTE
This counter is never reset during the lifetime of the acquisition channel.

Parameter Usage

Directive: To recover from this error, it is required to terminate the current acquisition sequence and restart a new one.

Parameter Values

Value	Description
0	No occurence *Minimum range value. Default value.*

4.12. Grabber Configuration Category

Parameters controlling the hardware resources specific to the grabber used by the channel

Connector

Connector used by the channel

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	SELECT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
682 << 14	Connector	MC_Connector

Parameter Description

The value of this parameter is entered at the channel creation by means of the Connector argument. The consistency of this parameter should be maintained channel-wide.

Parameter Values

A

DualBase

MC_Connector_A

DualBase

Description

B

DualBase

MC_Connector_B

DualBase

Description

M

Base Full FullXR

MC_Connector_M

Base Full FullXR

Description

The channel is linked to a Camera Link Medium, Full, or 10-tap configuration camera at both the Camera connector #1 and #2 or to a Camera Link Base configuration camera at the Camera connector #1.

ConnectLoc

Base DualBase Full FullXR

Connector location on the bracket where the relevant camera is connected

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	SELECT	Enumerated	Get Only

Num ID	String Identifier	C, C++ identifier
694 << 14	ConnectLoc	MC_ConnectLoc

Parameter Description

ConnectLoc is an informational parameter reflecting the argument entered by the application at the channel creation.

Parameter Values

UPPER

DualBase Full FullXR

MC_ConnectLoc_UPPER

Description
The channel uses a camera connected to the upper bracket connector.

LOWER

Base DualBase

MC_ConnectLoc_LOWER

Description
The channel uses a camera connected to the connector at the lower bracket position.

BOTH

Full FullXR

MC_ConnectLoc_BOTH

Description
The channel uses a camera connected to both (upper and lower) bracket connectors.

EqualizationLevel

FullXR

Channel link equalizer(s) level settings

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10667 << 14	EqualizationLevel	MC_EqualizationLevel

Parameter Description

The boards featuring ECCO+ are equipped with cable equalizers on the Camera Link receivers.

The equalizers compensate for the attenuation of the highest frequencies of the signal along the data lines of the Camera Link cable.

The equalizers provide four selectable equalization levels:

- OFF: 0 dB - The equalizers are turned off.
- LOW: The equalizers compensate for a cable attenuation of 4 dB at 1 GHz.
- MEDIUM: The equalizers compensate for a cable attenuation of 8 dB at 1 GHz.
- HIGH: The equalizers compensate for a cable attenuation of 16 dB at 1 GHz.

The cable attenuation is proportional to the cable length and to the cable attenuation characteristic. It can be estimated as follows:

attenuation [dB] = *cable_length* [m] x *cable_attenuation_characteristic* [dB/m]

NOTE
An AWG28 twisted pair, commonly found in Camera Link cable assemblies exhibits a typical *cable_attenuation_characteristic* of 1.4 dB/m

Parameter Usage

Relevance condition(s):

Condition: ECCO+ feature is available

Directive: Select the equalization level according to the actual cable attenuation.

Directive: For AWG28 cables of up to 1 meter, select OFF.

Directive: For AWG28 cables of 1 up to 4 meters, select LOW.

Directive: For AWG28 cables of 4 up to 8 meters, select MEDIUM.

Directive: For AWG28 cables of 8 up to 20 meters, select HIGH.

Parameter Values

OFF

FullXR

MC_EqualizationLevel_OFF

Description
Equalizers are turned OFF

LOW

FullXR

MC_EqualizationLevel_LOW

Description
Equalizers are turned ON with a low gain

MEDIUM

FullXR

MC_EqualizationLevel_MEDIUM

Description
Equalizers are turned ON with a medium gain

HIGH

FullXR

MC_EqualizationLevel_HIGH

Description
Equalizers are turned ON with a high gain

Default value.

PoCL_Mode

Base DualBase FullXR

PoCL control

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
9784 << 14	PoCL_Mode	MC_PoCL_Mode

Parameter Description

Enables/inhibits the PoCL controller to automatically detect a PoCL camera and activate the powering of the camera through the Camera Link cable.

NOTE
Any modification of this parameter is only effective after setting the MultiCam channel in the ready state.

Parameter Usage

Directive: Set the ChannelState parameter to value READY after any modification of PoCL_Mode.

Directive: To avoid unexpected activation of PoCL when the camera is not powered trough the Camera Link cable, set to OFF.

Parameter Values

AUTO

Base DualBase FullXR

MC_PoCL_Mode_AUTO
Description The PoCL controller(s) identifie(s) automatically the type of camera, and configures the camera power distribution accordingly.
Default value.

OFF

Base DualBase FullXR

MC_PoCL_Mode_OFF

Description
The camera detection is inhibited, and, if power is already applied, the PoCL controller turns off the power.

ECCO_PLLResetControl

Base DualBase Full FullXR

ECCO PLL reset control

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10570 << 14	ECCO_PLLResetControl	MC_ECCO_PLLResetControl

Parameter Description

Selects the method to reset the Phase Locked Loop (PLL) of the ECCO Camera Link receivers owned by the channel.

Parameter Usage

Relevance condition(s):

Condition: The ECCO feature is used. Namely, when BoardTopology is not equal to MONO_SLOW nor DUO_SLOW.

Directive: Euresys recommends the AUTOMATIC setting.

Parameter Values

AUTOMATIC

Base DualBase Full FullXR

MC_ECCO_PLLResetControl_AUTOMATIC
Description The reset of the PLL is automatically managed by the ECCO circuit.
Default value.

CHANNEL_ACTIVATION

Base DualBase Full FullXR

MC_ECCO_PLLResetControl_CHANNEL_ACTIVATION
Description The reset of the PLL is enforced at every channel activation.
Applicability condition(s) *Condition:* BoardTopology is not equal to MONO_SLOW nor DUO_SLOW

ECCO_SkewCompensation

Base DualBase Full FullXR

ECCO skew compensation control

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10568 << 14	ECCO_SkewCompensation	MC_ECCO_SkewCompensation

Parameter Description

Enable/disable the skew compensation function of the ECCO Camera Link receivers owned by the channel.

Parameter Usage

Relevance condition(s):

Condition: The ECCO feature is used. Namely, when BoardTopology is not equal to MONO_SLOW nor DUO_SLOW.

Directive: Euresys recommends to keep the de-skew function enabled. Disabling the de-skew function should be used for test purpose exclusively.

Parameter Values

ECCO_SkewCompensation_ON

Base DualBase Full FullXR

MC_ECCO_SkewCompensation_ECCO_SkewCompensation_ON
Description The skew compensation function is enabled
Default value.

ECCO_SkewCompensation_OFF

Base DualBase Full FullXR

MC_ECCO_SkewCompensation_ECCO_SkewCompensation_OFF

Description
The skew compensation function is disabled

Applicability condition(s)
Condition: BoardTopology is not equal to MONO_SLOW nor DUO_SLOW

FvalMin_Tk

Base DualBase Full FullXR

Camera Link FVAL digital filter configuration

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
10628 << 14	FvalMin_Tk	MC_FvalMin_Tk

Parameter Description

Configures the digital filter of the Camera Link FVAL input signal owned by the channel.

Parameter Usage

Directive: Euresys recommends keeping the filter setting to its default value.

Parameter Values

1

Base DualBase Full FullXR

MC_FvalMin_Tk_1
Description Does not filter FVAL high pulses; FVAL pulses as narrow as 1 clock period are considered as valid

3

Base DualBase Full FullXR

MC_FvalMin_Tk_3
Description Filter out FVAL high pulses narrower than 3 clock periods
Default value.

LvalMin_Tk

Base DualBase Full FullXR

LVAL digital filter configuration

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
10629 << 14	LvalMin_Tk	MC_LvalMin_Tk

Parameter Description

Configures the digital filter of the Camera Link LVAL input signal(s) owned by the channel.

Parameter Usage

Directive: Euresys recommends keeping the filter setting to its default value.

Parameter Values

1

Base DualBase Full FullXR

MC_LvalMin_Tk_1
Description Does not filter LVAL high pulses; LVAL pulses as narrow as 1 clock period are considered as valid

2

Base DualBase Full FullXR

MC_LvalMin_Tk_2
Description Filter out LVAL high pulses narrower than 2 clock periods
Default value.

PoCL_Status

Base DualBase FullXR

PoCL controller status

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
9941 << 14	PoCL_Status	MC_PoCL_Status

Parameter Description

Reports the status of the PoCL controller.

Parameter Usage

Directive: Check the status to troubleshoot inoperative PoCL.

Camera presence detection

To be detected, the attached camera must deliver a clock signal on Channel_Link_X.

Conventional camera detection

Possible causes explaining why the detected camera is identified as a non_PoCL camera are:

- The camera is not PoCL compliant
- The cable is not PoCL compliant

PoCL camera detection

To be detected as a PoCL camera, the camera and the cable must be PoCL compliant.For dual cable configurations (MEDIUM, FULL, 80-bit), the camera is declared PoCL compliant if at least one of the two PoCL controllers identifies a PoCL-compliant camera/cable combination.

Parameter Values

NO_CAMERA

Base DualBase FullXR

MC_PoCL_Status_NO_CAMERA
Description No camera detected.

CONVENTIONAL_CAMERA

Base DualBase FullXR

MC_PoCL_Status_CONVENTIONAL_CAMERA

Description
Conventional non-PoCL camera detected.

PoCL_CAMERA

Base DualBase FullXR

MC_PoCL_Status_PoCL_CAMERA

Description
PoCL camera detected.

MetadataInsertion

Base DualBase Full FullXR

Controls metadata insertion into the image

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10842 << 14	MetadataInsertion	MC_MetadataInsertion

Parameter Description

This enumerated parameter controls the insertion of metadata into the image.

Parameter Usage

The setting takes effect at the first channel activation.

Parameter Values

ENABLE

Base DualBase Full FullXR

MC_MetadataInsertion_ENABLE
Description Enable insertion of metadata into the image.

DISABLE

Base DualBase Full FullXR

MC_MetadataInsertion_DISABLE
Description Disable insertion of metadata into the image.
Default value.

MetadataContent

Base DualBase Full FullXR

Reports the metadata content configuration

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	EXPERT	Enumerated	Get Only

Num ID	String Identifier	C, C++ identifier
10849 << 14	MetadataContent	MC_MetadataContent

Parameter Description

This enumerated parameter reports the configuration of the metadata content.

Parameter Usage

The setting takes effect at the first channel activation.

Parameter Values

NONE

Base DualBase Full FullXR

MC_MetadataContent_NONE

Description
There are no metadata content.
This occurs when MetadataInsertion = DISABLE or when the camera interface configuration doesn't allow metadata insertion.

ONE_FIELD

Base DualBase Full FullXR

MC_MetadataContent_ONE_FIELD

Description
The metadata content includes one single field: the I/O state.

TWO_FIELD

Base DualBase Full FullXR

MC_MetadataContent_TWO_FIELD

Description
The metadata content includes two fields: the I/O state and the LVAL count.

THREE_FIELD

Base DualBase Full FullXR

MC_MetadataContent_THREE_FIELD

Description
The metadata content includes three fields: I/O state, LVAL count and encoder pulse count.

MetadataLocation

Base DualBase Full FullXR

Defines metadata location in the Camera Link data stream

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11055 << 14	MetadataLocation	MC_MetadataLocation

Parameter Description

This enumerated parameter controls the location of metadata in the Camera Link data stream.

Parameter Usage

Relevance condition(s):

Condition: The parameter is relevant only when the insertion of metadata is enabled: MetadataInsertion = ENABLE.

Parameter Values

LVALRISE

Base DualBase Full FullXR

MC_MetadataLocation_LVALRISE
Description The metadata content is located into all taps of the first Camera Link time slot following the rising edge of the Camera Link LVAL signal.

TAP1

Full FullXR

MC_MetadataLocation_TAP1

Description
The metadata content is inserted into the first tap during 10 consecutive Camera Link time slots following the rising edge of the Camera Link LVAL signal.

Applicability condition(s)
Condition: Imaging = LINE or TDI: The camera is a line-scan or line-scan TDI camera.
Condition: TapConfiguration = MEDIUM_4T8: The camera uses an Medium Camera Link configuration.
Condition: TapGeometry = 4X: The camera uses a four X-regions tap geometry.

TAP10

Full FullXR

MC_MetadataLocation_TAP10

Description
The metadata content is inserted into the 10th tap during 10 consecutive Camera Link time slots following the rising edge of the Camera Link LVAL signal.

Applicability condition(s)
Condition: Imaging = LINE or TDI: The camera is a line-scan or line-scan TDI camera.
Condition: TapConfiguration = DECA_10T8: The camera uses an 80-bit (10 taps of 8-bit) Camera Link configuration.
Condition: TapGeometry = 1X10: The camera uses a single X-region tap geometry.

MetadataGPPCInputLine

Full FullXR

GPPC main control

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11211 << 14	MetadataGPPCInputLine	MC_MetadataGPPCInputLine

Parameter Description

This enumerated parameter is the main control of the general purpose pulse counter.

Parameter Usage

Directive: Set to IIN1 to enable the general purpose pulse counter.

Parameter Values

NONE

Full FullXR

MC_MetadataGPPCInputLine_NONE
Description The GGPC is disabled. The counter has no input line!
Default value.

IIN1

Full FullXR

MC_MetadataGPPCInputLine_IIN1
Description The GGPC counts the rising edge events applied to the IIN1 isolated input line.

MetadataGPPCLocation

Full FullXR

GPPC metadata location in the Camera Link data stream

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11213 << 14	MetadataGPPCLocation	MC_MetadataGPPCLocation

Parameter Values

NONE

Full FullXR

MC_MetadataGPPCLocation_NONE

Description
The GPPC metadata is not inserted in the Camera Link data stream.

INSTEAD_LVALCNT

Full FullXR

MC_MetadataGPPCLocation_INSTEAD_LVALCNT

Description
The GPPC metadata replaces the LVAL Count metadata in the Camera Link data stream.

INSTEAD_QCNT

Full FullXR

MC_MetadataGPPCLocation_INSTEAD_QCNT

Description
The GPPC metadata replaces the Q Count metadata in the Camera Link data stream.

MetadataGPPCResetLine

Full FullXR

GPPC reset line control

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11214 << 14	MetadataGPPCResetLine	MC_MetadataGPPCResetLine

Parameter Values

NONE

Full FullXR

MC_MetadataGPPCResetLine_NONE
Description The GPPC has no reset input line.

IIN4

Full FullXR

MC_MetadataGPPCResetLine_IIN4
Description The GPPC resets when a high-level is applied to the IIN4 isolated input line.

MetadataSampleTime

Full FullXR

Metadata sample time selector

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Configuration	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11223 << 14	MetadataSampleTime	MC_MetadataSampleTime

Parameter Description

Defines the metadata sample time.

Parameter Usage

Prerequisite action(s):

Condition: BoardTopology = MONO_DECA

Parameter Values

LVALRISE

Full FullXR

MC_MetadataSampleTime_LVALRISE
Description Metadata are sampled on each rising edge of LVAL.
Default value.

EXPOSURE

Full FullXR

MC_MetadataSampleTime_EXPOSURE
Description Metadata are sampled on each "start of exposure" event.

4.13. Grabber Timing Category

Parameters controlling the hardware resources specific to the grabber used by the channel

GrabWindow

Method to define the grabbing window area

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Timing	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
683 << 14	GrabWindow	MC_GrabWindow

Parameter Description

This enumerated parameter selects the method defining the grabbing window area within the camera active area.

For area-scan cameras, the grabbing window area is inferred from the camera active rectangular window.

For line-scan cameras, the width of the grabbing window area is inferred from the camera active linear window.

Parameter Usage

By default, the grabbing window is the largest area achievable by the camera sensor.

Alternatively, using the MAN setting, the grabbing window can be reduced to a single rectangular area located anywhere in the camera active area.

Parameter Values

MAN

MC_GrabWindow_MAN
Description For area-scan cameras, the grabbing window area and location are defined by separate parameters: • Grabbing window width is defined by WindowX_Px. • Grabbing window height is defined by WindowY_Ln. • Grabbing window X-position offset is defined by OffsetX_Px. • Grabbing window Y-position offset is defined by OffsetY_Ln. For digital line-scan cameras, the grabbing window width and position are defined by separate parameters: • Grabbing window width is defined by WindowX_Px. • Grabbing window X-position offset is defined by OffsetX_Px.
Applicability condition(s) *Condition:* Line-scan or TDI line-scan cameras *Condition:* Area-scan cameras having a single region along the Y direction. For instance, the value is not applicable to cameras having a TapGeometry value suffixed _2YE.

WindowX_Px

Width of the grabbing window area

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Timing	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
826 << 14	WindowX_Px	MC_WindowX_Px

Parameter Description

This integer parameter reflects the width of the grabbing window area, expressed as a number of digitized pixels. The "get" value exactly reflects the actual window width. It may differ from the "set" value established by the user since MultiCam automatically corrects invalid values.

Parameter Usage

Relevance condition(s):

Condition: Manually defined grabbing window area (GrabWindow = MAN)

Prerequisite action(s):

Condition: Grabbing window definition method already selected through GrabWindow

Directive: Assigning a value smaller than Hactive_Px enables the image cropping feature.

Directive: The grabbing window area must be included entirely within the camera active area.

Parameter Values

Base DualBase Full FullXR

Value	Description
8	8 pixels *Minimum range value.*
Variable	Hactive_Px pixels *Maximum range value.*

WindowY_Ln

Height of the grabbing window area

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Timing	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
827 << 14	WindowY_Ln	MC_WindowY_Ln

Parameter Description

This integer parameter reflects the height of the grabbing window area, expressed as a number of lines.

The "get" value exactly reflects the actual window height. It may differ from the "set" value established by the user since MultiCam automatically corrects invalid values.

The parameter is available on all MultiCam products supporting area-scan cameras.

The parameter can be set when GrabWindow is set to MAN.

Parameter Usage

Relevance condition(s):

Condition: Manually defined grabbing window area (GrabWindow = MAN)

Condition: Area-scan camera (Imaging = AREA) having a single region along the Y direction (TapGeometry ≠ *_2YE)

Prerequisite action(s):

Condition: Grabbing window definition method already selected through GrabWindow

Directive: Assigning a value smaller than Vactive_Ln enables the image cropping feature.

Directive: The grabbing window area must be included entirely within the camera active area.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 line *Minimum range value.*
Variable	Vactive_Ln lines *Maximum range value.*

OffsetX_Px

Horizontal position offset of the grabbing window area in the camera active area

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Timing	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
825 << 14	OffsetX_Px	MC_OffsetX_Px

Parameter Description

This integer parameter reflects the horizontal position offset of the center of the grabbing window area relative to the center of the camera active area.

The "get" value exactly reflects the shifted amount. It may differ from the "set" value established by the user since MultiCam automatically corrects invalid values.

Parameter Usage

Relevance condition(s):

Condition: Manually defined grabbing window area (GrabWindow = MAN)

Prerequisite action(s):

Condition: Grabbing window definition method already selected through GrabWindow

Condition: Grabbing window height already set through WindowY_Ln

Directive: A value of zero means that the grabbing window area is horizontally centered on the Camera Active Area. Increasing the value shifts the grabbing window area in the right direction. Decreasing the value shifts the grabbing window area in the left direction.

Directive: The grabbing window area must be included entirely within the camera active area.

Parameter Values

Value	Description
Variable	{-(Hactive_Px - WindowX_Px) / 2} : Leftmost position within the camera active area *Minimum range value.*
Variable	{(Hactive_Px - WindowX_Px + 1) / 2} : Rightmost position within the camera active area *Maximum range value.*
0	The grabbing window area is horizontally centered on the grabbing window area *Default value.*

OffsetY_Ln

Vertical position offset of the grabbing window area in the camera active area.

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Timing	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
686 << 14	OffsetY_Ln	MC_OffsetY_Ln

Parameter Description

This integer parameter reflects the vertical position offset of the center of the Window Area relative to the center of the camera active area.

The "get" value exactly reflects the shifted amount. It may differ from the "set" value established by the user since MultiCam automatically corrects invalid values.

Parameter Usage

Relevance condition(s):

Condition: Manually defined grabbing window area (GrabWindow = MAN)

Condition: Area-scan camera (Imaging = AREA) having a single region along the Y direction (TapGeometry ≠ *_2YE)

Prerequisite action(s):

Condition: Grabbing window definition method already selected through GrabWindow

Condition: Grabbing window height already set through WindowY_Ln

Directive: Assigning a value of zero means that the grabbing window area is vertically centered on the Camera Active Area. Increasing the value shifts the grabbing window area in the downward direction. Decreasing the value shifts the grabbing window area in the upward direction.

Directive: The grabbing window area must be included entirely within the camera active area.

Parameter Values

Value	Description
	{- (Vactive_Ln - WindowY_Ln) / 2} : Uppermost position within the camera active area *Minimum range value.*
	{(Vactive_Ln - WindowY_Ln + 1) / 2} : Lowermost position within the camera active area *Maximum range value.*
0	The grabbing window area is vertically centered on the grabbing window area. *Default value.*

WindowOrgX_Px

Base DualBase Full FullXR

X-coordinate of the upper-left corner of the grabbing window area

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Timing	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
8761 << 14	WindowOrgX_Px	MC_WindowOrgX_Px

Parameter Description

This integer parameter reports the X-coordinate, expressed as a number of pixels, of the upper left corner of the grabbing window area.

Parameter Usage

Relevance condition(s):

Condition: Manually defined grabbing window area (GrabWindow = MAN)

Parameter Values

Value	Description
0	X-coordinate of the first column of the grabbing window area *Minimum range value.*
Variable	(Hactive_Px - WindowX_Px) *Maximum range value.*

WindowOrgY_Ln

Base DualBase Full FullXR

Y-coordinate of the upper-left corner of the grabbing window area

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Timing	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
8765 << 14	WindowOrgY_Ln	MC_WindowOrgY_Ln

Parameter Description

This integer parameter reports the Y-coordinate, expressed as a number of lines, of the upper left corner of the grabbing window area.

Parameter Usage

Relevance condition(s):

Condition: Manually defined grabbing window area (GrabWindow = MAN)

Condition: Area-scan camera (Imaging = AREA) having a single region along the Y direction (TapGeometry ≠ *_2YE)

Parameter Values

Value	Description
0	Y-coordinate of the first row of the grabbing window area *Minimum range value.*
Variable	(Vactive_Ln - WindowY_Ln) *Maximum range value.*

4.14. Grabber Conditioning Category

Parameters controlling the analog or digital conditioning features applied to the video signal processed by the grabber used by the channel

CFD_Mode

Base DualBase Full FullXR

Bayer decoding algorithm

Parameter Info

Class	Category	Level	Type	Access
Channel	Grabber Conditioning	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
4645 << 14	CFD_Mode	MC_CFD_Mode

Parameter Values

ADVANCED

Base DualBase Full FullXR

MC_CFD_Mode_ADVANCED

Description
Bayer decoding algorithm using a 3x3 interpolation and a median filter.

LEGACY

Base DualBase Full FullXR

MC_CFD_Mode_LEGACY

Description
Bayer decoding algorithm using a 3x3 interpolation identical to eVision Bayer decoding function.

4.15. White Balance Operator Category

Parameters controlling the white balance operator used by the channel

WBO_Mode

Base DualBase Full FullXR

Operating mode of the white balance operator

Parameter Info

Class	Category	Level	Type	Access
Channel	White Balance Operator	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
4715 << 14	WBO_Mode	MC_WBO_Mode

Parameter Description

This enumerated parameter determines the operating mode of the White Balance Operator within a MultiCam acquisition sequence.

Parameter Usage

Relevance condition(s):

Condition: The camera is a color camera (Spectrum=COLOR).

Condition: The acquisition channel delivers Y and/or RGB pixel data (ColorFormat ≠ BAYER*)

Parameter Values

NONE

Base DualBase Full FullXR

MC_WBO_Mode_NONE
Description When WBO_Mode is set to NONE, the White Balance Operator is disabled; the gain corrections are not applied.
Default value.

ONCE

Base DualBase Full FullXR

MC_WBO_Mode_ONCE

Description
When WBO_Mode is set to ONCE, the image color balancing gains are automatically computed during the initial acquisition phase of every MultiCam acquisition sequence within the AWB_AREA defined by parameters WBO_OrgX, WBO_OrgY, WBO_Width, and WBO_Height.
The parameters WBO_GainR, WBO_GainG, and WBO_GainR are automatically set to the respective computed gain values.

Base DualBase Full FullXR

Description
The White Balance Operator is disabled at the begin of the sequence and remains disabled until the occurrence of the first MC_SIG_SURFACE_PROCESSING signal. The first delivered image is never color balanced; subsequent images remain partially or entirely unbalance until the White Balance Operator is configured.

MANUAL

Base DualBase Full FullXR

MC_WBO_Mode_MANUAL

Description
When WBO_Mode is set to MANUAL, the image color balance is performed with gains specified by parameters WBO_GainR, WBO_GainG and WBO_GainB.

WBO_GainR

Base DualBase Full FullXR

White balance correction factor for the red color component

Parameter Info

Class	Category	Level	Type	Access
Channel	White Balance Operator	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
4717 << 14	WBO_GainR	MC_WBO_GainR

Parameter Description

This integer parameter represents the correction factor applied by the White Balance Operator to the red color component.

The parameter values are expressed in 1/1000th. For instance a value of 1234 corresponds to a correction factor of 1.234.

Parameter Usage

Relevance condition(s):

Condition: Manually defined WBO gains (WBO_Mode = MANUAL)

Prerequisite action(s):

Condition: The WBO operation mode is already selected through WBO_Mode

Parameter Values

Value	Description
1000	The gain correction factor is 1 *Minimum range value. Default value.*
10000	The gain correction factor is 10 *Maximum range value.*

WBO_GainG

Base DualBase Full FullXR

White balance correction factor for the green color component

Parameter Info

Class	Category	Level	Type	Access
Channel	White Balance Operator	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
4719 << 14	WBO_GainG	MC_WBO_GainG

Parameter Description

This integer parameter represents the correction factor applied by the White Balance Operator to the green color component.

The parameter values are expressed in 1/1000th. For instance a value of 1234 corresponds to a correction factor of 1.234.

Parameter Usage

Relevance condition(s):

Condition: Manually defined WBO gains (WBO_Mode = MANUAL)

Prerequisite action(s):

Condition: The WBO operation mode is already selected through WBO_Mode

Parameter Values

Value	Description
1000	The gain correction factor is 1 *Minimum range value. Default value.*
10000	The gain correction factor is 10 *Maximum range value.*

WBO_GainB

Base DualBase Full FullXR

White balance correction factor for the blue color component

Parameter Info

Class	Category	Level	Type	Access
Channel	White Balance Operator	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
4720 << 14	WBO_GainB	MC_WBO_GainB

Parameter Description

This integer parameter represents the correction factor applied by the White Balance Operator to the blue color component.

The parameter values are expressed in 1/1000th. For instance a value of 1234 corresponds to a correction factor of 1.234.

Parameter Usage

Relevance condition(s):

Condition: Manually defined WBO gains (WBO_Mode = MANUAL)

Prerequisite action(s):

Condition: The WBO operation mode is already selected through WBO_Mode

Parameter Values

Value	Description
1000	The gain correction factor is 1 *Minimum range value. Default value.*
10000	The gain correction factor is 10 *Maximum range value.*

WBO_Width

Base DualBase Full FullXR

Width of the Automatic White Balance Area

Parameter Info

Class	Category	Level	Type	Access
Channel	White Balance Operator	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
5456 << 14	WBO_Width	MC_WBO_Width

Parameter Description

This integer parameter represents the width, expressed as a number of pixels, of the rectangular region within the camera active area that is used by the Automatic White Balance feature to compute the white balance correction factors.

Parameter Usage

Relevance condition(s):

Condition: Automatically defined WBO gains (WBO_Mode = ONCE or CONTINUOUS)

Prerequisite action(s):

Condition: The WBO operation mode is already selected through WBO_Mode

Directive: The AWB_AREA must include at least 256 pixels.

Directive: The AWB_AREA must include at least 32 columns of pixels.

Directive: The AWB_AREA must be included entirely within the camera active area.

Directive: The AWB_AREA must be included entirely within the grabbing window area.

Parameter Values

Base DualBase Full FullXR

Value	Description
32	32 pixels *Minimum range value.*
Variable	(Hactive_Px - WBO_OrgX) *Maximum range value. Default value.*

WBO_Height

Base DualBase Full FullXR

Height of the Automatic White Balance Area

Parameter Info

Class	Category	Level	Type	Access
Channel	White Balance Operator	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
5459 << 14	WBO_Height	MC_WBO_Height

Parameter Description

This integer parameter represents the height, expressed as a number of lines, of the rectangular region within the camera active area that is used by the Automatic White Balance feature to compute the white balance correction factors.

Parameter Usage

Relevance condition(s):

Condition: Automatically defined WBO gains (WBO_Mode = ONCE or CONTINUOUS)

Prerequisite action(s):

Condition: The WBO operation mode is already selected through WBO_Mode

Directive: The AWB_AREA must include at least 256 pixels.

Directive: The AWB_AREA must include at least 1 line of pixels.

Directive: The AWB_AREA must be included entirely within the camera active area.

Directive: The AWB_AREA must be included entirely within the grabbing window area.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	1 line *Minimum range value.*
Variable	(Vactive_Ln - WBO_OrgY) *Maximum range value. Default value.*

WBO_OrgX

Base DualBase Full FullXR

X-coordinate of the upper-left corner of the AWB_AREA

Parameter Info

Class	Category	Level	Type	Access
Channel	White Balance Operator	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
5449 << 14	WBO_OrgX	MC_WBO_OrgX

Parameter Description

This integer parameter represents the X-coordinate, expressed as a number of pixels, of the upper left corner of arectangular region within the camera active area that is used by the Automatic White Balance feature to compute the white balance correction factors.

Parameter Usage

Relevance condition(s):

Condition: Automatically defined WBO gains (WBO_Mode = ONCE or CONTINUOUS)

Prerequisite action(s):

Condition: The WBO operation mode is already selected through WBO_Mode

Directive: The AWB_AREA must include at least 256 pixels.

Directive: The AWB_AREA must include at least 32 columns of pixels.

Directive: The AWB_AREA must be included entirely within the camera active area.

Directive: The AWB_AREA must be included entirely within the grabbing window area.

Parameter Values

Value	Description
0	Leftmost column of the grabbing window area *Minimum range value.*
Variable	Hactive_Px - 32 *Maximum range value.*

WBO_OrgY

Base DualBase Full FullXR

Y-coordinate of the upper-left corner of the AWB_AREA

Parameter Info

Class	Category	Level	Type	Access
Channel	White Balance Operator	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
5452 << 14	WBO_OrgY	MC_WBO_OrgY

Parameter Description

This integer parameter represents the Y-coordinate, expressed as a number of lines, of the upper left corner of a rectangular region within the camera active area that is used by the Automatic White Balance feature to compute the white balance correction factors.

Parameter Usage

Relevance condition(s):

Condition: Automatically defined WBO gains (WBO_Mode = ONCE or CONTINUOUS)

Prerequisite action(s):

Condition: The WBO operation mode is already selected through WBO_Mode

Directive: The AWB_AREA must include at least 256 pixels.

Directive: The AWB_AREA must include at least 1 line of pixels.

Directive: The AWB_AREA must be included entirely within the camera active area.

Directive: The AWB_AREA must be included entirely within the grabbing window area.

Parameter Values

Value	Description
0	Uppermost row of the grabbing window area *Minimum range value.*
Variable	(Vactive_Ln - 1) *Maximum range value.*

WBO_Status

Base DualBase Full FullXR

Status of the automatic white balance learning block

Parameter Info

Class	Category	Level	Type	Access
Channel	White Balance Operator	EXPERT	Enumerated	Get Only

Num ID	String Identifier	C, C++ identifier
8940 << 14	WBO_Status	MC_WBO_Status

Parameter Description

This enumerated parameter shows the result status of the automatic white balance computation.

Parameter Usage

Relevance condition(s):

Condition: Automatically defined WBO gains (WBO_Mode = ONCE or CONTINUOUS)

Prerequisite action(s):

Condition: The WBO operation mode is already selected through WBO_Mode

Condition: At least one acquiistion phase already completed.

Parameter Values

OK

Base DualBase Full FullXR

MC_WBO_Status_OK
Description The automatic white balance learning block succeeds to balance the color. The white balance color gain settings are updated.

NOT_OK

Base DualBase Full FullXR

MC_WBO_Status_NOT_OK
Description The automatic white balance learning block fails to balance the color. The white balance color gain settings are not updated.

4.16. Look-up Tables Category

Parameters controlling the look-up-table operator used by the channel

LUT_Method

Base DualBase Full FullXR

LUT construction method

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
4969 << 14	LUT_Method	MC_LUT_Method

Parameter Description

Once the LUT has been defined through any of the construction methods, the values table can not be read back.

Parameter Values

RESPONSE_CONTROL

Base DualBase Full FullXR

MC_LUT_Method_RESPONSE_CONTROL

Description
The LUT relevant control parameters are LUT_Contrast , LUT_Visibility , LUT_Brightness and LUT_Negative .

EMPHASIS

Base DualBase Full FullXR

MC_LUT_Method_EMPHASIS

Description
The LUT relevant control parameters are LUT_Emphasis and LUT_Negative .

THRESHOLD

Base DualBase Full FullXR

MC_LUT_Method_THRESHOLD

Description
The LUT relevant control parameters are LUT_SlicingLevel , LUT_SlicingBand , LUT_LightResponse , LUT_BandResponse and LUT_DarkResponse .

TABLE

Base DualBase Full FullXR

MC_LUT_Method_TABLE

Description
The LUT table is defined through the LUT_Table parameter.

LUT_StoreIndex

Base DualBase Full FullXR

Index in the board memory of the LUT to store.

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
2957 << 14	LUT_StoreIndex	MC_LUT_StoreIndex

Parameter Description

Setting this parameter stores a pre-defined LUT in the board memory. Multiple LUTs can be stored together, and the index defines the LUT place inside the memory.

LUT_UseIndex

Base DualBase Full FullXR

Index in the board memory of the LUT to activate.

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
2956 << 14	LUT_UseIndex	MC_LUT_UseIndex

Parameter Description

Setting this parameter activates immediately the defined LUT stored in the board memory.

LUT_Contrast

Base DualBase Full FullXR

Contrast factor for a LUT defined through the Response Control method.

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Float collection	Set and Get

Num ID	String Identifier	C, C++ identifier
2952 << 14	LUT_Contrast	MC_LUT_Contrast

Parameter Description

If the application is managing **monochrome formats**, this parameter is a collection of 4 elements with:

- Element 0 not relevant.
- Element 1 not relevant.
- Element 2 not relevant.
- Element 3 associated with the image.

If the application is managing **planar** or **packed color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 not relevant.

If the application is managing **combined planar color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 associated with the Y or luminance (gray level) component of the image.

LUT_Brightness

Base DualBase Full FullXR

Brightness factor for a LUT defined through the Response Control method.

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Float collection	Set and Get

Num ID	String Identifier	C, C++ identifier
2953 << 14	LUT_Brightness	MC_LUT_Brightness

Parameter Description

If the application is managing **monochrome formats**, this parameter is a collection of 4 elements with:

- Element 0 not relevant.
- Element 1 not relevant.
- Element 2 not relevant.
- Element 3 associated with the image.

If the application is managing **planar** or **packed color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 not relevant.

If the application is managing **combined planar color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 associated with the Y or luminance (gray level) component of the image.

LUT_Visibility

Base DualBase Full FullXR

Visibility factor for a LUT defined through the Response Control method.

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Float collection	Set and Get

Num ID	String Identifier	C, C++ identifier
2954 << 14	LUT_Visibility	MC_LUT_Visibility

Parameter Description

If the application is managing **monochrome formats**, this parameter is a collection of 4 elements with:

- Element 0 not relevant.
- Element 1 not relevant.
- Element 2 not relevant.
- Element 3 associated with the image.

If the application is managing **planar** or **packed color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 not relevant.

If the application is managing **combined planar color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 associated with the Y or luminance (gray level) component of the image.

LUT_Negative

Base DualBase Full FullXR

Visibility factor for a LUT defined through the Response Control or the Emphasis method.

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Enumerated collection	Set and Get

Num ID	String Identifier	C, C++ identifier
2955 << 14	LUT_Negative	MC_LUT_Negative

Parameter Description

If the application is managing **monochrome formats**, this parameter is a collection of 4 elements with:

- Element 0 not relevant.
- Element 1 not relevant.
- Element 2 not relevant.
- Element 3 associated with the image.

If the application is managing **planar** or **packed color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 not relevant.

If the application is managing **combined planar color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 associated with the Y or luminance (gray level) component of the image.

Parameter Values

LUT_Emphasis

Base DualBase Full FullXR

Emphasis factor for a LUT defined through the Emphasis method.

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Float collection	Set and Get

Num ID	String Identifier	C, C++ identifier
4970 << 14	LUT_Emphasis	MC_LUT_Emphasis

Parameter Description

If the application is managing **monochrome formats**, this parameter is a collection of 4 elements with:

- Element 0 not relevant.
- Element 1 not relevant.
- Element 2 not relevant.
- Element 3 associated with the image.

If the application is managing **planar** or **packed color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 not relevant.

If the application is managing **combined planar color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 associated with the Y or luminance (gray level) component of the image.

LUT_SlicingLevel

Base DualBase Full FullXR

Chooses the level of slicing for a LUT defined through the Threshold method.

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Float collection	Set and Get

Num ID	String Identifier	C, C++ identifier
4971 << 14	LUT_SlicingLevel	MC_LUT_SlicingLevel

Parameter Description

If the application is managing **monochrome formats**, this parameter is a collection of 4 elements with:

- Element 0 not relevant.
- Element 1 not relevant.
- Element 2 not relevant.
- Element 3 associated with the image.

If the application is managing **planar** or **packed color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 not relevant.

If the application is managing **combined planar color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 associated with the Y or luminance (gray level) component of the image.

LUT_SlicingBand

Base DualBase Full FullXR

Band width of slicing for a LUT defined through the Threshold method.

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Float collection	Set and Get

Num ID	String Identifier	C, C++ identifier
4972 << 14	LUT_SlicingBand	MC_LUT_SlicingBand

Parameter Description

If the application is managing **monochrome formats**, this parameter is a collection of 4 elements with:

- Element 0 not relevant.
- Element 1 not relevant.
- Element 2 not relevant.
- Element 3 associated with the image.

If the application is managing **planar** or **packed color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 not relevant.

If the application is managing **combined planar color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 associated with the Y or luminance (gray level) component of the image.

LUT_LightResponse

Base DualBase Full FullXR

Response in the light part for a LUT defined through the Threshold method.

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Float collection	Set and Get

Num ID	String Identifier	C, C++ identifier
4973 << 14	LUT_LightResponse	MC_LUT_LightResponse

Parameter Description

If the application is managing **monochrome formats**, this parameter is a collection of 4 elements with:

- Element 0 not relevant.
- Element 1 not relevant.
- Element 2 not relevant.
- Element 3 associated with the image.

If the application is managing **planar** or **packed color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 not relevant.

If the application is managing **combined planar color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 associated with the Y or luminance (gray level) component of the image.

LUT_BandResponse

Base DualBase Full FullXR

Response in the middle part for a LUT defined through the Threshold method.

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Float collection	Set and Get

Num ID	String Identifier	C, C++ identifier
4974 << 14	LUT_BandResponse	MC_LUT_BandResponse

Parameter Description

If the application is managing **monochrome formats**, this parameter is a collection of 4 elements with:

- Element 0 not relevant.
- Element 1 not relevant.
- Element 2 not relevant.
- Element 3 associated with the image.

If the application is managing **planar** or **packed color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 not relevant.

If the application is managing **combined planar color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 associated with the Y or luminance (gray level) component of the image.

LUT_DarkResponse

Base DualBase Full FullXR

Response in the dark part for a LUT defined through the Threshold method.

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Float collection	Set and Get

Num ID	String Identifier	C, C++ identifier
4975 << 14	LUT_DarkResponse	MC_LUT_DarkResponse

Parameter Description

If the application is managing **monochrome formats**, this parameter is a collection of 4 elements with:

- Element 0 not relevant.
- Element 1 not relevant.
- Element 2 not relevant.
- Element 3 associated with the image.

If the application is managing **planar** or **packed color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 not relevant.

If the application is managing **combined planar color formats**, this parameter is a collection of 4 elements with:

- Element 0 associated with the R or red component of the image.
- Element 1 associated with the G or green component of the image.
- Element 2 associated with the B or blue component of the image.
- Element 3 associated with the Y or luminance (gray level) component of the image.

LUT_InDataWidth

Base DualBase Full FullXR

Digital data width of the LUT input

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
3017 << 14	LUT_InDataWidth	MC_LUT_InDataWidth

Parameter Description

Getting this parameter returns the number of significant data bits applied at the input of every LUT transformer.

Parameter Values

Base DualBase Full FullXR

Value	Description
16	16 bits

LUT_OutDataWidth

Base DualBase Full FullXR

Digital data width of the LUT output

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
3018 << 14	LUT_OutDataWidth	MC_LUT_OutDataWidth

Parameter Description

Getting this parameter returns the number of significant data bits delivered by every LUT transformer.

Parameter Values

Base DualBase Full FullXR

Value	Description
16	16 bits

LUT_Table

Base DualBase Full FullXR

Manually specifies a LUT defined through the Table method.

Parameter Info

Class	Category	Level	Type	Access
Channel	Look-up Tables	EXPERT	Instance	Set and Get

Num ID	String Identifier	C, C++ identifier
2951 << 14	LUT_Table	MC_LUT_Table

Parameter Description

Once the LUT has been defined through any method, the values table can not be read back.

4.17. Board Linkage Category

Parameters providing several methods to designate one of the frame grabber inside the system as the channel host

BoardName

Name of the board linked to the channel

Parameter Info

Class	Category	Level	Type	Access
Channel	Board Linkage	SELECT	String	Set and Get

Num ID	String Identifier	C, C++ identifier
2 << 14	BoardName	MC_BoardName

Parameter Description

This parameter provides a method to designate a particular board where the channel should find its grabber resources.

The designation is based on the name given to a board. The name is a string of maximum 16 ASCII characters.

DriverIndex

Board locator in the list returned by the driver

Parameter Info

Class	Category	Level	Type	Access
Channel	Board Linkage	SELECT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
0 << 14	DriverIndex	MC_DriverIndex

Parameter Description

This parameter provides a method to designate a particular board where the channel should find its grabber resources.

The designation is based on the board location in the list returned by the driver. The set of MultiCam compliant boards are assigned a set of consecutive integer numbers starting at 0. The indexing order is system dependent.

Setting this parameter links the board having the specified driver index to the channel.

Setting the parameter to an index larger than or equal to the number of MultiCam boards results in the MC_NO_BOARD_FOUND error.

PCIPosition

Board locator in the list of PCI slots

Parameter Info

Class	Category	Level	Type	Access
Channel	Board Linkage	SELECT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
1 << 14	PCIPosition	MC_PCIPosition

Parameter Description

This parameter provides a method to designate a particular board where the channel should find its grabber resources. Setting this parameter links the board inserted in the specified PCI slot to the channel

The designation is based on the number associated to a PCI slot. This number is assigned by the operating system in a non-predictable way, but remains consistent for a given configuration in a given system.

BoardIdentifier

Identifier of the board linked to the channel, made by the combination of its type and serial number

Parameter Info

Class	Category	Level	Type	Access
Channel	Board Linkage	SELECT	String	Set and Get

Num ID	String Identifier	C, C++ identifier
3 << 14	BoardIdentifier	MC_BoardIdentifier

Parameter Description

This parameter provides a method to designate a particular board where the channel should find its grabber resources.

The designation is based on the board type and its serial number, providing a unique way to designate a Euresys product.

The board identifier is an ASCII character string resulting from the concatenation of the board type and the serial number with an intervening underscore. The serial number is a 6-digit string made of characters 0 to 9.

4.18. Cluster Category

Parameters defining the destination surface cluster owned by the channel

Cluster

Set of surfaces associated to a channel

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	SELECT	Instance collection	Set and Get

Num ID	String Identifier	C, C++ identifier
12 << 14	Cluster	MC_Cluster

Parameter Description

This parameter gives access to the list of handles of the surfaces belonging to the destination cluster.

A cluster is a set of surfaces having compatible characteristics, but different locations. All surfaces belonging to a cluster should be able to accept images coming from the same source through a given channel.

The idea behind the clusters is the capability to easily implement advanced destination structures such as double, triple or rotating image buffers.

Surface to Cluster Assignment

A surface can be assigned to several clusters provided that:

- The clusters belong to channels defined within the same application.
- The channels address the same board.

The maximum number of surfaces assigned to a channel is 4096, and the maximum number of surfaces instantiated within an application is 4096.

Currently, the number of surfaces that can be handled by a board may be less than the maximum, depending on the hardware capabilities and characteristics of the acquisition surface.

ImageSizeX

Horizontal size of the transferred images, expressed as a number of columns

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	ADJUST	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
523 << 14	ImageSizeX	MC_ImageSizeX

Parameter Description

This parameter can be set only with Picolo boards.

It exposes the result of any condition adjustment that could affect the image width during the acquisition process.

The surface in the destination cluster will receive an image, the width of which is that number of columns.

In case of area-scan cameras, the size of the destination surface matches the size of the acquired frame.

In case of line-scan cameras, the size of the destination surface matches the size of the acquired page.

The horizontal size of the image is scaled to the defined ImageSizeX number of pixels per line.

ImageSizeY

Vertical size of the transferred images, expressed as a number of lines

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	ADJUST	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
524 << 14	ImageSizeY	MC_ImageSizeY

Parameter Description

This parameter can be set only with Picolo boards.

It exposes the result of any condition adjustment that could affect the image height during the acquisition process.

The surface in the destination cluster will receive an image the height of which is that number of lines.

In case of area-scan cameras, the size of the destination surface matches the size of the acquired frame.

In case of line-scan cameras, the size of the destination surface matches the size of the acquired page.

The vertical size of the image is scaled to the defined ImageSizeY number of lines.

ImageFlipX

Horizontal mirroring effect

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1340 << 14	ImageFlipX	MC_ImageFlipX

Parameter Description

The horizontal mirroring effect can be thought as turning the image around a vertical axis (first column becomes last column).

Parameter Values

OFF

MC_ImageFlipX_OFF
Description No horizontal mirroring effect.

ON

Base DualBase Full FullXR

MC_ImageFlipX_ON
Description Horizontal mirror applied.

ImageFlipY

Vertical mirroring effect

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
525 << 14	ImageFlipY	MC_ImageFlipY

Parameter Description

The vertical mirroring effect can be thought as turning the image around a horizontal axis (first line becomes last line).

Parameter Values

OFF

MC_ImageFlipY_OFF
Description No vertical mirroring effect.

ON

MC_ImageFlipY_ON
Description Vertical mirror applied.

ColorFormat

Color format

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
2224 << 14	ColorFormat	MC_ColorFormat

Parameter Description

This parameter summarizes all the properties describing how the frame grabber stores pixel data in the destination surface.

For a complete description of pixel storage formats, see "MultiCam Storage Formats" on page 512.

Parameter Values

Y8

Base DualBase Full FullXR

MC_ColorFormat_Y8
Description 8-bit monochrome pixel data; aligned to byte boundaries

Y10

Base DualBase Full FullXR

MC_ColorFormat_Y10
Description 10-bit monochrome pixel data justified to lsb; 6 padding bits of'0' for alignment to 16-bit

Y10P

Full FullXR

MC_ColorFormat_Y10P
Description 10-bit monochrome pixel data; no padding bits; packed storage (8 pixels are stored in 10 bytes)

Y12

Base DualBase Full FullXR

MC_ColorFormat_Y12

Description
12-bit monochrome pixel data justified to lsb; 4 padding bits of'0' for alignment to 16-bit

Y14

Base DualBase Full FullXR

MC_ColorFormat_Y14

Description
14-bit monochrome pixel data justified to lsb; 2 padding bits of'0' for alignment to 16-bit

Y16

Base DualBase Full FullXR

MC_ColorFormat_Y16

Description
16-bit monochrome pixel data; aligned to 16-bit boundaries

BAYER8

Base DualBase Full FullXR

MC_ColorFormat_BAYER8

Description
8-bit BAYER component data; aligned to byte boundaries

BAYER10

Base DualBase Full FullXR

MC_ColorFormat_BAYER10

Description
10-bit BAYER component data justified to lsb; 6 padding bits of'0' for alignment to 16-bit

BAYER12

Base DualBase Full FullXR

MC_ColorFormat_BAYER12

Description
12-bit BAYER component data justified to lsb; 4 padding bits of'0' for alignment to 16-bit

BAYER14

Base DualBase Full FullXR

MC_ColorFormat_BAYER14

Description
14-bit BAYER component data justified to lsb; 2 padding bits of'0' for alignment to 16-bit

BAYER16

Base DualBase Full FullXR

MC_ColorFormat_BAYER16

Description
16-bit BAYER component data; aligned to 16-bit boundaries

RGB24

Base DualBase Full FullXR

MC_ColorFormat_RGB24

Description
3x 8-bit packed color components data; each component is aligned to byte boundaries

RGB32

Base DualBase Full FullXR

MC_ColorFormat_RGB32

Description
4x 8-bit packed color components data; each component is aligned to byte boundaries

ARGB32

Base DualBase Full FullXR

MC_ColorFormat_ARGB32

Description
4x 8-bit packed color components data; each component is aligned to byte boundaries

RGB30P

Full FullXR

MC_ColorFormat_RGB30P

Description
3x 10-bit packed color components data; no padding bits; packed storage (8 pixels are stored in 30 bytes)

RGBI40P

Full FullXR

MC_ColorFormat_RGBI40P

Description
4x 10-bit packed color components data; no padding bits; packed storage (8 pixels are stored in 30 bytes)

RGB24PL

Base DualBase Full FullXR

MC_ColorFormat_RGB24PL

Description
3 planes of 8-bit color components data; each component is aligned to byte boundaries

Description
Each pixel color is stored using RGB24PL system.

RGB30PL

Base DualBase Full FullXR

MC_ColorFormat_RGB30PL

Description
3 planes of 10-bit color components data; each component is justified to lsb and padded with 6 bits of '0' for alignment to 16-bit

RGB36PL

Base DualBase Full FullXR

MC_ColorFormat_RGB36PL

Description
3 planes of 12-bit color components data; each component is justified to lsb and padded with 4 bits of '0' for alignment to 16-bit

RGB48PL

Base DualBase Full FullXR

MC_ColorFormat_RGB48PL

Description
Each pixel color is stored using RGB48PL system. In this storage format, the least 6 significant bits of the pixel value are 0.

RedBlueSwap

Base DualBase Full FullXR

Controls the swapping of the red and blue color components

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
11052 << 14	RedBlueSwap	MC_RedBlueSwap

Parameter Description

This enumerated parameter controls the swapping of the red and blue color components when acquiring color packed image data from RGB color cameras.

Parameter Usage

Relevance condition(s):

Condition: Parallel RGB camera (delivers 8-bit, 10-bit or 12-bit R, G, and B color components in parallel).

Condition: RGB color packed pixel format.

Prerequisite action(s):

Condition: Spectrum must be set to Color.

Condition: ColorMethod must be set RGB.

Condition: TapConfiguration must be set to BASE_1T24, MEDIUM_1T30, MEDIUM_1T36, MEDIUM_2T24 or DECA_3T24.

Condition: ColorFormat must be set to RGB24 or RGB32.

Parameter Values

ENABLE

Base DualBase Full FullXR

MC_RedBlueSwap_ENABLE

Description
The frame grabber swaps the Red and Blue components of the Camera Link RGB pixel data.

NOTE
This corresponds to the behaviour of MultiCam prior to Release 6.9.8.

Default value.

DISABLE

Base DualBase Full FullXR

MC_RedBlueSwap_DISABLE

Description
The frame grabber keeps the pixel component order of the Camera Link RGB pixel data.

ColorComponentsOrder

Base DualBase Full FullXR

Color components order

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	EXPERT	Enumerated	Get Only

Num ID	String Identifier	C, C++ identifier
11054 << 14	ColorComponentsOrder	MC_ColorComponentsOrder

Parameter Description

This enumerated parameter reports the color components order of RGB packed pixel formats.

Parameter Usage

Relevance condition(s):

Condition: Parallel RGB camera (delivers R, G, and B color components in parallel).

Condition: RGB color packed pixel format.

Prerequisite action(s):

Condition: Spectrum must be set to Color.

Condition: ColorMethod must be set RGB.

Condition: TapConfiguration must be set to BASE_1T24, MEDIUM_1T30, MEDIUM_1T36, MEDIUM_2T24 or DECA_3T24.

Condition: ColorFormat must be set to RGB24 or RGB32.

Parameter Values

RGB

Base DualBase Full FullXR

MC_ColorComponentsOrder_RGB
Description The color components order is RGB.

BGR

Base DualBase Full FullXR

MC_ColorComponentsOrder_BGR

Description
The color components order is BGR.

ImagePlaneCount

Number of image planes

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	ADJUST	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
1718 << 14	ImagePlaneCount	MC_ImagePlaneCount

Parameter Description

MultiCam creates the surfaces and automatically allocates the memory buffers, if not done by the application. The following channel parameters configure the automatic allocation: BufferSize , BufferPitch , ImagePlaneCount and SurfaceCount . MultiCam decides the adequate number of surfaces for the selected acquisition mode.

This parameter indicates the number of planes required by the frame grabber to store the pixel data.

The channel cannot be activated if all surfaces do not meet this requirement.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	Single-plane surface
3	Three-plane surface

BufferSize

Recommended size (in bytes) for the image buffer(s)

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	ADJUST	Integer collection	Get Only

Num ID	String Identifier	C, C++ identifier
3333 << 14	BufferSize	MC_BufferSize

Parameter Description

MultiCam creates the surfaces and automatically allocates the memory buffers, if not done by the application. The following channel parameters configure the automatic allocation: BufferSize , BufferPitch , ImagePlaneCount and SurfaceCount . MultiCam decides the adequate number of surfaces for the selected acquisition mode.

This parameter is expressed as a number of bytes.

It provides the buffer size needed to contain one image produced by the channel.

If ImagePlaneCount> 1, the channel produces a "multi-plane" image. In this case, one must allocate ImagePlaneCount buffers.

Each buffer size is given in the BufferSize collection members.

For instance, if ImagePlaneCount = 3, allocate 3 buffers.

- Buffer 1 size is indicated by BufferSize [0].
- Buffer 2 size is indicated by BufferSize [1].
- Buffer 3 size is indicated by BufferSize [2].

For more information about access to integer collections, refer to Parameters.

SurfaceIndex

Index of the next acquisition surface to fill

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
17 << 14	SurfaceIndex	MC_SurfaceIndex

Parameter Description

Getting this parameter gives access to the index of the lastly or currently written surface. This surface is in the FILLING state, as defined by SurfaceState , or got most recently the FILLED state.

Setting this parameter allows the selection of a surface to be used by the next acquisition phase. The target surface must be in the FREE state.

The value is the zero-based index of the surface in the cluster.

This parameter selects the strategy to be exercised by the capture controller.

SurfaceCount

Number of surfaces in the channel

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
82 << 14	SurfaceCount	MC_SurfaceCount

Parameter Description

MultiCam creates the surfaces and automatically allocates the memory buffers, if not done by the application. The following channel parameters configure the automatic allocation: BufferSize , BufferPitch , ImagePlaneCount and SurfaceCount . MultiCam decides the adequate number of surfaces for the selected acquisition mode.

Getting SurfaceCount indicates the number of surfaces in the channel. The user may change SurfaceCount to another value before channel activation.

LineIndex

Base DualBase Full FullXR

Index of the written line

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
16 << 14	LineIndex	MC_LineIndex

Parameter Description

This parameter gives access to the index of the line currently written into the FILLING surface, as defined by SurfaceState .

Parameter Values

Base DualBase Full FullXR

Value	Description
0	Firstly written line *Minimum range value.*

ImageColorRegistration

Base DualBase Full FullXR

Alignment of Bayer pattern filter over acquired surface

Parameter Info

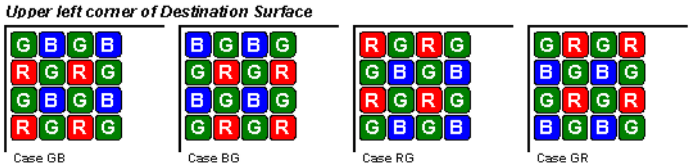
Class	Category	Level	Type	Access
Channel	Cluster	EXPERT	Enumerated	Get Only

Num ID	String Identifier	C, C++ identifier
1274 << 14	ImageColorRegistration	MC_ImageColorRegistration

Parameter Description

This parameter indicates how the Bayer pattern filter covers the image acquired in the destination surface. It applies when Spectrum is COLOR and ColorMethod is BAYER.

It is automatically set according to the value of ColorRegistration and according to the setting of the grabbing window.



Upper left corner of destination surface

Parameter Values

GB

Base DualBase Full FullXR

MC_ImageColorRegistration_GB
Description The first two pixels are green and blue.

BG

Base DualBase Full FullXR

MC_ImageColorRegistration_BG
Description The first two pixels are blue and green.

RG

Base DualBase Full FullXR

MC_ImageColorRegistration_RG

Description
The first two pixels are red and green.

GR

Base DualBase Full FullXR

MC_ImageColorRegistration_GR

Description
The first two pixels are green and red.

SurfacePlaneName

Base DualBase Full FullXR

Image component type stored for each plane

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	EXPERT	Enumerated collection	Get Only

Num ID	String Identifier	C, C++ identifier
4876 << 14	SurfacePlaneName	MC_SurfacePlaneName

Parameter Description

For a complete description of pixel storage formats, see "MultiCam Storage Formats" on page 512.

Parameter Values

UNUSED

Base DualBase Full FullXR

MC_SurfacePlaneName_UNUSED

Description
The plane does not exist.

Y

Base DualBase Full FullXR

MC_SurfacePlaneName_Y

Description
The plane holds the luminance component of the image.

YUV

Base DualBase Full FullXR

MC_SurfacePlaneName_YUV

Description
The plane holds the image in a YUV color packed format.

R

Base DualBase Full FullXR

MC_SurfacePlaneName_R

Description
The plane holds the red component of the image in a color planar format.

G

Base DualBase Full FullXR

MC_SurfacePlaneName_G

Description
The plane holds the green component of the image in a color planar format.

B

Base DualBase Full FullXR

MC_SurfacePlaneName_B

Description
The plane holds the blue component of the image in a color planar format.

RGB

Base DualBase Full FullXR

MC_SurfacePlaneName_RGB

Description
The plane holds the image in a RGB color packed format.

YRGB

Base DualBase Full FullXR

MC_SurfacePlaneName_YRGB

Description
The plane holds the image in a combined luminance and RGB color packed format.

MinBufferPitch

Minimum size to contain one line of the image plane, expressed as a number of bytes.

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	EXPERT	Integer collection	Get Only

Num ID	String Identifier	C, C++ identifier
3335 << 14	MinBufferPitch	MC_MinBufferPitch

Parameter Description

This parameter indicates the minimal size required to contain one line of the image plane produced by the channel. The channel cannot be activated if all surfaces do not meet this requirement.

The line pitch size is defined by parameter BufferPitch .

The dimension of this collection parameter is specified by ImagePlaneCount . The assignment of the planes is returned by SurfacePlaneName . For a complete description of pixel storage formats, see "MultiCam Storage Formats" on page 512.

Parameter Values

Value	Description
4	4 bytes *Minimum range value.*
32768	32,768 bytes *Maximum range value.*

BufferPitch

Size required to contain one line of the plane, expressed in bytes

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	EXPERT	Integer collection	Set and Get

Num ID	String Identifier	C, C++ identifier
3336 << 14	BufferPitch	MC_BufferPitch

Parameter Description

MultiCam creates the surfaces and automatically allocates the memory buffers, if not done by the application. The following channel parameters configure the automatic allocation: BufferSize , BufferPitch , ImagePlaneCount and SurfaceCount . MultiCam decides the adequate number of surfaces for the selected acquisition mode.

Getting this parameter gives the minimum size (in bytes) required to contain one line of the plane produced by the channel.

Setting this parameter defines the desired line pitch. If allowed, this value will be used in the computation of other "Cluster Category " on page 414

The minimum value is reported by parameter MinBufferPitch .

The dimension of this collection parameter is specified by ImagePlaneCount . The assignment of the planes is returned by SurfacePlaneName . For a complete description of pixel storage formats, see "MultiCam Storage Formats" on page 512.

MinBufferSize

Minimal size required to contain the image plane, expressed as a number of bytes.

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	EXPERT	Integer collection	Get Only

Num ID	String Identifier	C, C++ identifier
3334 << 14	MinBufferSize	MC_MinBufferSize

Parameter Description

This parameter indicates the absolute minimal buffer size accepted by the channel.

If the size of one or more surface buffers is below the corresponding MinBufferSize , the channel will report an error at activation and image acquisition will not be possible.

The dimension of this collection parameter is specified by ImagePlaneCount . The assignment of the planes is returned by SurfacePlaneName . For a complete description of pixel storage formats, see "MultiCam Storage Formats" on page 512.

SurfaceAllocation

Memory allocation method of MultiCam surfaces

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10092 << 14	SurfaceAllocation	MC_SurfaceAllocation

Parameter Description

MultiCam sets automatically this parameter to the right value, so there should be no need to modify it.

Parameter Values

MaxFillingSurfaces

Base DualBase Full FullXR

Filling surfaces control

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
10712 << 14	MaxFillingSurfaces	MC_MaxFillingSurfaces

Parameter Description

This parameter specifies the operation of the cluster mechanism regarding the number of surfaces it is allowed to put in the FILLING state.

Parameter Usage

Prerequisite action(s):

Condition: The parameter must be set prior to the channel activation, i.e. when ChannelState = IDLE

Directive: Allocate a sufficient amount of surfaces and manage the surfaces such that the cluster mechanism maintains a sufficient amount of surfaces in the MC_SurfaceState_FILLING state to cover the largest system interrupt latencies.

Parameter Values

MINIMUM

Base DualBase Full FullXR

MC_MaxFillingSurfaces_MINIMUM

Description
The cluster mechanism is allowed to put **only one** surface in the FILLING state at a time.

MAXIMUM

Base DualBase Full FullXR

MC_MaxFillingSurfaces_MAXIMUM

Description
The cluster mechanism is allowed to put up to 512 surfaces in the FILLING state at a time.

Default value.

FifoOrdering

Base DualBase Full FullXR

Video lines reordering control

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1719 << 14	FifoOrdering	MC_FifoOrdering

Parameter Description

This parameter controls the reordering of video lines in the acquisition buffer.

For more information, refer to Video Lines Reordering in the Grablink Functional Guide.

Parameter Usage

Directive: For area-scan cameras having a *_2YE TapGeometry value, MultiCam automatically sets the parameter value to DUALYEND.

Directive: For other cameras, MultiCam automatically sets the parameter value to PROGRESSIVE.

Directive: For multi-color multi-spectral line-scan cameras delivering video data lines by block of N lines, the user may set the parameter value to NYTAP to group lines by color planes.

Parameter Values

PROGRESSIVE

Base DualBase Full FullXR

MC_FifoOrdering_PROGRESSIVE

Description
The frame grabber doesn't reorder video lines. This is the default value except for area-scan cameras having a *_2YE TapGeometry value.

DUALYEND

Base DualBase Full FullXR

MC_FifoOrdering_DUALYEND

Description
The frame grabber re-orders the video-lines according to the DUALYEND reordering scheme. This is the default value for area-scan cameras having a *_2YE TapGeometry value.

NYTAP

Base DualBase Full FullXR

MC_FifoOrdering_NYTAP

Description
The frame grabber re-orders the video-lines according to the NYTAP reordering scheme.

PENTAYTAP

Base DualBase Full FullXR

MC_FifoOrdering_PENTAYTAP

Description
The frame grabber re-orders the video-lines according to the PENTAYTAP reordering scheme.

Description
This value is kept for backward compatibility.

FifoOrderingYTapCount

Base DualBase Full FullXR

Number of taps in the Y-direction for video lines reordering

Parameter Info

Class	Category	Level	Type	Access
Channel	Cluster	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
11225 << 14	FifoOrderingYTapCount	MC_FifoOrderingYTapCount

Parameter Description

This parameter allows choosing the number of wanted Y taps (planes) when FifoOrdering=NYTAP.

For other values of the FifoOrdering parameter, it takes an appropriate default value and changing it has no effect.

Parameter Values

Base DualBase Full FullXR

Value	Description
1	Single tap *Condition:* FifoOrdering=NYTAP *Minimum range value. Default value.*
Variable	ImageSizeY taps *Condition:* FifoOrdering=NYTAP *Maximum range value.*

Base DualBase Full FullXR

Value	Description
1	Single tap *Condition:* FifoOrdering=PROGRESSIVE *Default value.*
2	Two taps *Condition:* FifoOrdering=DUALYEND *Default value.*
5	Five taps *Condition:* FifoOrdering=PENTAYTAP *Default value.*

4.19. Channel Management Category

Parameters controlling state information of the channel

ChannelState

State of the channel

Parameter Info

Class	Category	Level	Type	Access
Channel	Channel Management	SELECT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
15 << 14	ChannelState	MC_ChannelState

Parameter Description

Refer to "Automatic Switching" on page 495.

Parameter Values

IDLE

MC_ChannelState_IDLE

Description
The channel owns the grabber at this moment but does not lock it.

Description
Sets the channel's state to IDLE or READY.

ACTIVE

MC_ChannelState_ACTIVE

Description
The channel uses the grabber.

ORPHAN

Base DualBase Full FullXR

MC_ChannelState_ORPHAN

Description
The channel has no grabber.

READY

Base DualBase Full FullXR

MC_ChannelState_READY

Description
The channel locks the grabber and is ready to start an acquisition sequence.

FREE

Base DualBase Full FullXR

MC_ChannelState_FREE

Description
Try to set the channel's state to ORPHAN.

CallbackPriority

Priority of the callback thread

Parameter Info

Class	Category	Level	Type	Access
Channel	Channel Management	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
100 << 14	CallbackPriority	MC_CallbackPriority

Parameter Description

Registering a callback function results into the creation in the application process of a separate thread dedicated to the callback function. This thread is maintained idle until a signal occurs. This parameter can be used to select the priority of this callback thread.

Refer to Registration of Callback Function and "Callback Signaling" on page 497 for information on MultiCam callbacks.

Parameter Values

LOWEST

Base DualBase Full FullXR

MC_CallbackPriority_LOWEST
Description

BELOW_NORMAL

Base DualBase Full FullXR

MC_CallbackPriority_BELOW_NORMAL
Description

NORMAL

Base DualBase Full FullXR

MC_CallbackPriority_NORMAL
Description

ABOVE_NORMAL

Base DualBase Full FullXR

MC_CallbackPriority_ABOVE_NORMAL

Description

HIGHEST

Base DualBase Full FullXR

MC_CallbackPriority_HIGHEST

Description

TIME_CRITICAL

Base DualBase Full FullXR

MC_CallbackPriority_TIME_CRITICAL

Description

4.20. Signaling Category

Parameters controlling signaling information of the channel

SignalEnable

Selection of callback or waiting signals

Parameter Info

Class	Category	Level	Type	Access
Channel	Signaling	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
24 << 14	SignalEnable	MC_SignalEnable

Parameter Description

This collection parameter selects the MultiCam signals able to call a callback function or to trigger a waiting function.

For more information, refer to "Enabling Signals" on page 507.

Parameter Values

ON

MC_SignalEnable_ON
Description The signal is included in the selection.

OFF

MC_SignalEnable_OFF
Description The signal is not included in the selection.

AFTER_EAS

MC_SignalEnable_AFTER_EAS
Description The signal is disabled until the end of acquisition sequence.

SignalEvent

Operating system events associated with a MultiCam signals

Parameter Info

Class	Category	Level	Type	Access
Channel	Signaling	EXPERT	Enumerated	Get Only

Num ID	String Identifier	C, C++ identifier
25 << 14	SignalEvent	MC_SignalEvent

Parameter Description

This collection parameter holds operating system handles to event objects that are signaled when MultiCam signals occur.

SignalHandling

Signaling method to use when MultiCam signal appears

Parameter Info

Class	Category	Level	Type	Access
Channel	Signaling	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
74 << 14	SignalHandling	MC_SignalHandling

Parameter Description

This parameter selects which signaling method is used when a particular MultiCam signal appears.

If an application needs to use one method for signals of a particular type and another method for other signals, it must define this parameter for all concerned signals. If only one signaling method is used for all types of signals, this parameter does not have to be set.

If the setting of SignalHandling :s is CALLBACK_SIGNALING, each signal of type s will cause the callback function to be called. The MultiCam wait function McWaitSignal called for signal s will not be released upon occurrence of a signal of type s. Likewise, the OS event linked to signal s (SignalEvent :s) will not be signaled.

If the setting of SignalHandling :s is WAITING_SIGNALING, each signal of type s will release the MultiCam wait function McWaitSignal called for signal s. The callback function will not be called upon occurrence of a signal of type s. Likewise, the OS event linked to signal s (SignalEvent :s) will not be signaled.

If the setting of SignalHandling :s is OS_EVENT_SIGNALING, each signal of type s will cause the corresponding OS event (SignalEvent :s) to be signaled. The callback function will not be called upon occurrence of a signal of type s. Likewise, the MultiCam wait function McWaitSignal called for signal s will not be released.

Parameter Values

ANY

MC_SignalHandling_ANY
Description No signaling method has been selected.

CALLBACK_SIGNALING

MC_SignalHandling_CALLBACK_SIGNALING
Description The callback signaling method is used.

WAITING_SIGNALING

MC_SignalHandling_WAITING_SIGNALING

Description
The waiting signaling method is used.

OS_EVENT_SIGNALING

MC_SignalHandling_OS_EVENT_SIGNALING

Description
The OS event signaling method is used.

GenerateSignal

Signal generation mode

Parameter Info

Class	Category	Level	Type	Access
Channel	Signaling	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
73 << 14	GenerateSignal	MC_GenerateSignal

Parameter Description

This parameter is used to choose between two possible modes of signal generation.

By default, each MultiCam event produces a signal (if the corresponding signal is enabled). If a signal cannot be generated when the event occurs, the signal is queued.

In the other signal generation mode, the signals are not queued. MultiCam only keeps information about the latest event of each type.

Parameter Values

EACH_EVENT

MC_GenerateSignal_EACH_EVENT
Description Each MultiCam event produces a signal. If necessary, the signals are queued by MultiCam.
Default value.

LATEST_EVENT

MC_GenerateSignal_LATEST_EVENT
Description The signals are not queued by MultiCam.

4.21. Exception Management Category

Parameters controlling the exception situations encountered by the channel

AcquisitionCleanup

Base DualBase Full FullXR

Filtering of spoiled images

Parameter Info

Class	Category	Level	Type	Access
Channel	Exception Management	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
3024 << 14	AcquisitionCleanup	MC_AcquisitionCleanup

Parameter Description

Some acquired images may be spoiled, due to FIFO overruns for example. This parameter allows not to transfer these images to the application.

Parameter Values

ENABLED

Base DualBase Full FullXR

MC_AcquisitionCleanup_ENABLED

Description
The spoiled images are not signalled. The corresponding surfaces (SurfaceState) are immediately set to FREE.

DISABLED

Base DualBase Full FullXR

MC_AcquisitionCleanup_DISABLED

Description
The spoiled images are managed the same way as accurate images.

AcqTimeout_ms

Configuration of the acquisition timeout

Parameter Info

Class	Category	Level	Type	Access
Channel	Exception Management	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
56 << 14	Timeout	MC_AcqTimeout_ms

Parameter Description

This parameter controls the acquisition timeout:

- The timeout duration can be configured in steps of 1 millisecond.
- The timeout function can be disabled.

NOTE
The string identifier differs from the parameter name for backward compatibility reasons.

Parameter Usage

Directive: The parameter must be set prior to the activation of the channel.

Parameter Values

Base DualBase Full FullXR

Value	Description
10	10 milliseconds timeout duration *Minimum range value.*
10000	10,000 milliseconds (= 10 seconds) timeout duration *Default value.*
1000000	1,000,000 milliseconds (= 16 minutes and 40 seconds) timeout duration *Maximum range value.*
MC_INDETERMINATE (-1)	Disabled timeout function (= infinite duration)

OverrunCount

Base DualBase Full FullXR

Counter of overrun occurrences

Parameter Info

Class	Category	Level	Type	Access
Channel	Exception Management	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
46 << 14	OverrunCount	MC_OverrunCount

Parameter Description

This parameter returns the number of overrun occurrences since the creation of the channel. It is incremented each time a transfer overrun occurs. It may be initialized at any time by setting its value.

An overrun is an exception condition occurring when the data transfer between the frame grabber and the host computer saturates the PCI bus.

TriggerSkipHold

Base DualBase Full FullXR

Protection method of trigger

Parameter Info

Class	Category	Level	Type	Access
Channel	Exception Management	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
1309 << 14	TriggerSkipHold	MC_TriggerSkipHold

Parameter Description

When the trigger frequency is faster than allowable, the requested trigger is either skipped or held until the end of the current acquisition phase.

When TriggerSkipHold is set to HOLD, only the last trigger is maintained in the queue, even if many triggers appeared.

Parameter Values

SKIP

Base DualBase Full FullXR

MC_TriggerSkipHold_SKIP

Description
A trigger event is ignored if occurring while a previous trigger is already being treated.

HOLD

Base DualBase Full FullXR

MC_TriggerSkipHold_HOLD

Description
A trigger event is hold if occurring when a previous trigger is already being treated. The end of the current trigger event will be chained with the "hold" trigger event.

LineTriggerViolation

Base DualBase Full FullXR

Counter of line trigger violation occurrences

Parameter Info

Class	Category	Level	Type	Access
Channel	Exception Management	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
57 << 14	LineTriggerViolation	MC_LineTriggerViolation

Parameter Description

Getting this parameter reports the current value of the line trigger violation counter.

Setting this parameter sets an initial value for the counter.

The counter increments when one of the following events occurs:

- Line trigger occurs too quickly. (When the camera and illumination controller is not yet able to start a new cycle.)
- An excessive backward motion distance. (Occurs only when BackwardMotionCancellation is set to COMPENSATE.)
- An excessive period of time between consecutive line triggers. (Occurs only when LineRateMode is set to CONVERT.)

Parameter Usage

Relevance condition(s):

Condition: Line-scan cameras.

FrameTriggerViolation

Counter of frame trigger violation occurrences

Parameter Info

Class	Category	Level	Type	Access
Channel	Exception Management	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
58 << 14	FrameTriggerViolation	MC_FrameTriggerViolation

Parameter Description

This parameter increases when frame trigger violations occur. It is not incremented each time a frame trigger violation occurs. At least, it will be incremented once per page (line-scan) or per frame (area-scan).

Setting this parameter sets an initial value for the counter.

In case of area-scan operation, a frame trigger violation occurs when the frame trigger pulses occur too quickly. In case of line-scan operation, the rule applies to the page trigger pulses.

5. Surface Class

What Is a Surface?

The surface is a container where a 2D image can be stored. In most situations, the surface is a buffer in the host memory. Other types of surfaces may be defined, such as the hardware frame buffer located inside a frame grabber. In the particular case of a line-scan camera, the surface can be used as a circular buffer. This implies that, although the surface is 2D-limited, the incoming data flow is continuous and virtually unlimited.

Regarding the acquisition process, the surface is the destination where the grabbed images from the cameras are recorded. The overall goal of the MultiCam driver is to provide flexible channels to route images coming from a camera towards a specified surface.

Surface creation

The Surface class groups all MultiCam parameters dedicated to the definition of memory buffers for image or data storage. A Surface object is an instance of the Surface class represented by a dedicated set of such parameters that uniquely describe the surface.

Several surfaces can exist simultaneously. A process called "surface creation" is applied to define a new surface. A created surface is entirely characterized by a corresponding instance of the Surface class in the MultiCam environment.

Surfaces can be deleted by their owning application with an appropriate API function.

5.1. Surface Specification Category

Parameters specifying the static attributes of the surface

SurfaceSize

Size of the surface for one plane, expressed in bytes

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Specification	ADJUST	Integer collection	Set and Get

Num ID	String Identifier	C, C++ identifier
27 << 14	SurfaceSize	MC_SurfaceSize

Parameter Description

This parameter should be defined large enough to hold the intended image in the adequate format.

For backward compatibility, when it is used as an integer, it gives access to the first plane.

SurfaceAddr

Address of the surface for one plane, or list of addresses of the surface planes

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Specification	ADJUST	Pointer collection	Set and Get

Num ID	String Identifier	C, C++ identifier
28 << 14	SurfaceAddr	MC_SurfaceAddr

Parameter Description

If PlaneCount > 1, this parameter is a collection of the starting addresses for every plane constituting the surface.

NOTE

For backward compatibility, it is still possible to use an integer collection instead of a pointer collection. When it is used as an integer, it gives access to the first plane.

SurfacePitch

Pitch of the surfaces, expressed in bytes

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Specification	ADJUST	Integer collection	Set and Get

Num ID	String Identifier	C, C++ identifier
29 << 14	SurfacePitch	MC_SurfacePitch

Parameter Description

This parameter declares the pitch between vertically adjacent pixels of the surface for one plane.

For backward compatibility, when it is used as an integer, it gives access to the first plane.

PlaneCount

Number of planes in the surface

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Specification	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
30 << 14	PlaneCount	MC_PlaneCount

Parameter Description

Usually, the number of planes in the surface is 1. But some image formats, such as planar representation of RGB data, require more than one plane. PlaneCount is changed automatically when the collection parameter SurfaceAddr is set. All planes constituting the surface have a similar structure, but the starting address of each plane is different.

SurfaceContext

Placeholder for a pointer-precision user-defined value associated with this surface

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Specification	EXPERT	Pointer	Set and Get

Num ID	String Identifier	C, C++ identifier
32 << 14	SurfaceContext	MC_SurfaceContext

Parameter Description

This parameter provides a convenient way of declaring a user-defined context associated with a MultiCam surface using a pointer value. This context can be easily retrieved from the surface handle in a callback or waiting function.

For backward compatibility, it is still possible to use a 32-bit integer value instead of a pointer.

SurfaceSizeX

Horizontal image size in pixels

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Specification	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
84 << 14	SurfaceSizeX	MC_SurfaceSizeX

Parameter Description

This parameter holds the horizontal image size expressed in pixels. It is deduced from the ImageSizeX of the channel.

NOTE

This parameter access is "Get Only" when the surface belongs to the cluster of surfaces associated with a channel in the ACTIVE state.

SurfaceSizeY

Vertical image size in pixels

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Specification	EXPERT	Integer	Set and Get

Num ID	String Identifier	C, C++ identifier
85 << 14	SurfaceSizeY	MC_SurfaceSizeY

Parameter Description

This parameter holds the vertical image size expressed in pixels. It is deduced from the ImageSizeY of the channel.

NOTE

This parameter access is "Get Only" when the surface belongs to the cluster of surfaces associated with a channel in the ACTIVE state.

SurfaceColorFormat

Internal organization of pixels of the surface

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Specification	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
86 << 14	SurfaceColorFormat	MC_SurfaceColorFormat

Parameter Description

This parameter holds the internal pixel organization. It is deduced from the ColorFormat of the channel.

NOTE
This parameter access is "Get Only" when the surface belongs to the cluster of surfaces associated with a channel in the ACTIVE state.

Parameter Values

The possible values are described in the "ColorFormat" on page 420 parameter.

SurfaceColorRegistration

Base DualBase Full FullXR

Alignment of the color pattern filter over the camera window

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Specification	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
93 << 14	SurfaceColorRegistration	MC_SurfaceColorRegistration

Parameter Description

This parameter indicates how the Bayer pattern filter covers the camera active window. It is deduced from the ColorRegistration of the channel.

NOTE

This parameter access is "Get Only" when the surface belongs to the cluster of surfaces associated with a channel in the ACTIVE state.

Parameter Values

The possible values are described in the "ColorRegistration" on page 168 parameter.

SurfaceColorComponentsOrder

Base DualBase Full FullXR

Color components order of RGB packed pixel formats

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Specification	EXPERT	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
94 << 14	SurfaceColorComponentsOrder	MC_SurfaceColorComponentsOrder

Parameter Description

This parameter reports the color components order of RGB packed pixel formats. It is deduced from the ColorComponentsOrder of the channel.

NOTE
This parameter access is "Get Only" when the surface belongs to the cluster of surfaces associated with a channel in the ACTIVE state.

Parameter Values

The possible values are described in the "ColorComponentsOrder" on page 426 parameter.

5.2. Surface Dynamics Category

Parameters specifying the dynamic attributes of the surface

SurfaceState

State of the surface

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Dynamics	ADJUST	Enumerated	Set and Get

Num ID	String Identifier	C, C++ identifier
31 << 14	SurfaceState	MC_SurfaceState

Parameter Description

Get or set the current state of the surface.

Parameter Values

FREE

MC_SurfaceState_FREE
Description The surface is able to receive image data from the grabber.

FILLING

MC_SurfaceState_FILLING
Description The surface is currently receiving or ready to receive image data from the grabber.

FILLED

MC_SurfaceState_FILLED
Description The surface has finished receiving image data from the grabber, and thus is ready for processing

PROCESSING

MC_SurfaceState_PROCESSING
Description The surface is being processed by the host processor.

RESERVED

MC_SurfaceState_RESERVED

Description
The surface is removed from the standard state transition.

LastInSequence

Base DualBase Full FullXR

Last acquired surface in an acquisition sequence

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Dynamics	ADJUST	Enumerated	Get Only

Num ID	String Identifier	C, C++ identifier
92 << 14	LastInSequence	MC_LastInSequence

Parameter Description

This parameter indicates whether a surface is the last one in an acquisition sequence.

Parameter Values

TRUE

Base DualBase Full FullXR

MC_LastInSequence_TRUE
Description The surface is the last one of an acquisition sequence.

FALSE

Base DualBase Full FullXR

MC_LastInSequence_FALSE
Description The surface is not the last one of an acquisition sequence.

FillCount

Number of bytes written by the acquisition

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Dynamics	EXPERT	Integer collection	Get Only

Num ID	String Identifier	C, C++ identifier
43 << 14	FillCount	MC_FillCount

Parameter Description

This parameter holds the number of bytes actually written into the surface by the acquisition in this plane.

TimeCode

Internal numbering of surface during acquisition sequence

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Dynamics	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
72 << 14	TimeCode	MC_TimeCode

Parameter Description

The timecode is the order in a sequence. The first acquisition of a sequence is numbered 0, the second 1, and so on. The last acquired surface has the number SeqLength_Ph-1. If an acquisition happens but is not signaled to the application (for example, when no surface is available: cluster unavailable), the timecode is still incremented. The timecode is reset to 0 at each new sequence (channel state -> ACTIVE).

TimeAnsi

ANSI time of surface filled event

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Dynamics	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
76 << 14	TimeAnsi	MC_TimeAnsi

Parameter Description

This parameter represents the number of seconds elapsed since midnight (00:00:00), January 1, 1970, coordinated universal time (UTC), according to the system clock when the surface is filled.

TimeStamp_us

Time of surface filled event

Parameter Info

Class	Category	Level	Type	Access
Surface	Surface Dynamics	EXPERT	Integer	Get Only

Num ID	String Identifier	C, C++ identifier
77 << 14	TimeStamp_us	MC_TimeStamp_us

Parameter Description

This parameter represents the number of microseconds elapsed since midnight (00:00:00), January 1, 1970, coordinated universal time (UTC), according to the system clock when the surface is filled.

This parameter is a 64-bit integer.

NOTE

For backward compatibility, this parameter may still be a collection of two 32-bit integers; one for the low part and one for the high part.

6. Annex

6.1. MultiCam Acquisition Principles

Refer to *D405EN MultiCam Acquisition Principles* PDF document

6.2. TapConfiguration Glossary

Naming Convention

A tap configuration is designated by:

<Config>_<TapCount>T<BitDepth>(B<TimeSlots>)

<Config>

Designates the Camera Link configuration as follows:

Camera Link Configuration name	<Config> value
Lite	LITE
Base	BASE
Medium	MEDIUM
Full	FULL
72-bit	DECA
80-bit	DECA

<TapCount>

Total number of pixel taps. Values range: 1 to 10.

<BitDepth>

Number of bits per tap. Values list: {8, 10, 12, 14, 16, 24, 30, 36, 42, 48}.

<TimeSlots>

Number of consecutive time slots required to transfer one pixel data. Values list: {2, 3}

The field and the letter B are omitted when a single time slot is sufficient to deliver all the pixel data.

Examples

BASE_1T8: Base Camera Link configuration, 1 tap, 8-bit pixel data

BASE_1T24: Base Camera Link configuration, 1 tap, 24-bit pixel data (likely RGB)

DECA_8T10: 80-bit Camera Link configuration, 8 taps, 10-bit pixel data

DECA_8T30B3: 80-bit Camera Link configuration, 8 taps, 30-bit pixel data (likely RGB), 3 time slots

6.3. TapGeometry Glossary

Definitions

Adjacent taps

Two taps are adjacent when the extracted pixels are adjacent on the same row or on the same column.

Region

A rectangular area of adjacent pixels that are transferred in a raster-scan order through one or multiple adjacent taps.

Tap

One pixel stream output port of the camera that delivers one pixel every clock cycle.

Tap Geometrical Properties

A tap is characterized by the following properties:

XStart: X-position of the first extracted pixel of a camera readout cycle

XEnd: X-position of the last extracted pixel of a camera readout cycle

YStart: Y-position of the first extracted pixel of a camera readout cycle

YEnd: Y-position of the last extracted pixel of a camera readout cycle

YStep: the difference of Y-position between consecutive rows of pixels; it is positive when Y-position values are increasing (top to bottom); it is negative otherwise.

X-Position: the pixel column number in the (non-flipped) image; column 1 is the leftmost column; column W is the rightmost column of an image having a width of W pixels.

Y-Position: the pixel row number in the (non-flipped) image; row 1 is the topmost row; row H is the bottommost row of an image having a height of H pixels.

TapGeometry Values Syntax

There are two variants of the syntax:

1. For cameras delivering two or more rows of pixels every camera readout cycle:

   <TapGeometryX>_<TapGeometryY>

2. For cameras delivering only one row of pixels every camera, e.g. single line line-scan cameras:

   <TapGeometryX>

TapGeometryX Syntax

<TapGeometryX> describes the geometrical organization of the taps along one row of the image. It is built as follows:

<XRegions>X(<XTaps>)(<ExtX)>)

- <XRegions>: an integer declaring the number of regions encountered across one image row (= the X-direction or the horizontal direction). Possible values are 1, 2, 3, 4, 6, 8, and 10.
- <XTaps>: an integer declaring the number of consecutive pixels along one region row that are extracted simultaneously.
  Possible values are 1, 2, 3, 4, 8, and 10.
  The field is omitted when <XTaps> is 1.
- <ExtX>: a letter declaring the relative location of the pixels extractors across one row of the image.
  - This field is omitted when all pixel extractors are at the left of each region.
  - Letter E indicates that pixel extractors are at both ends of the image row.
  - Letter M indicates that pixel extractors are at middle of the image row.
  - Letter R indicates that the pixel extractors are all at the right of each region

TapGeometryY Syntax

<TapGeometryY> describes the geometrical organization of the taps along one column of the image. It is built as follows:

<YRegions>Y(<YTaps>)(<ExtY)>)

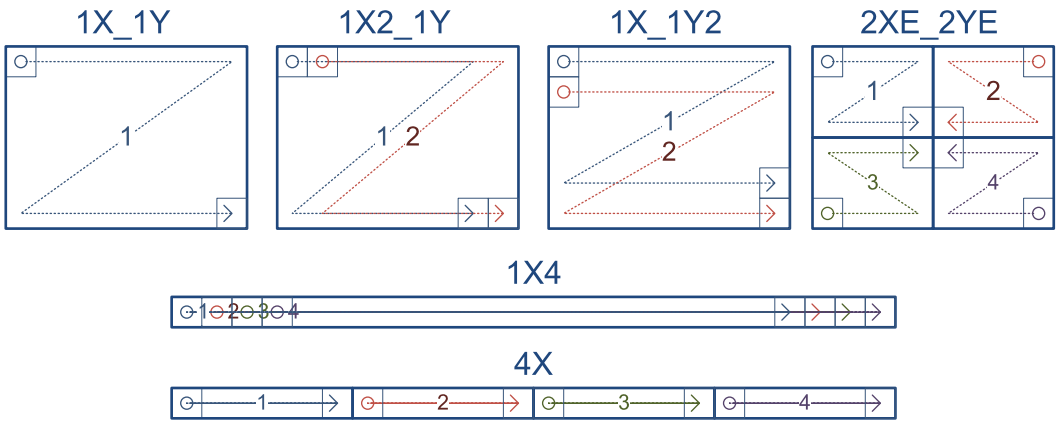
<YRegions>: an integer declaring the number of regions encountered across vertical direction. Possible values are 1 and 2.

<YTaps>: an integer declaring the number of consecutive pixels along one region column that are extracted simultaneously.
Possible values are 1 and 2. The field is omitted when YTaps is 1.

<ExtY>: a letter declaring the relative location of the pixels extractors across one column of the image.

- This field is omitted when all pixel extractors are at the top of each region.
- Letter E indicates that pixel extractors are at both ends of the image column.

TapGeometry Values Examples



1X_1Y designates the tap geometry of a single-tap camera having 1 region across the X-direction and 1 region across the Y direction.

The pixels are delivered one at a time on a single tap beginning with the leftmost pixel of the top row, scanning progressively all the rows of the image one by one, and ending with the rightmost pixel of the bottom row.

1X2_1Y designates the tap geometry of a two-tap camera having 1 region across the X-direction and 1 region across the Y direction.

The pixels are delivered two at a time on two taps beginning with the two leftmost pixels of the top row, scanning progressively all the rows of the image one by one, and ending with the two rightmost pixels of the bottom row.

1X_1Y2 designates the tap geometry of a two-tap camera having 1 region across the X-direction and 1 region across the Y direction.

The pixels are delivered two at a time on two taps beginning with the two uppermost pixels of the first column , scanning progressively all the rows of the image two by two, and ending with the two lowermost pixels of the rightmost column.

2XE_2YE designates the tap geometry of a four-tap camera having 2 regions across the X-direction and 2 regions across the Y direction.

The pixels are delivered four at a time on four taps. Each region delivers its pixels on a single-tap using a specific scanning scheme:

The pixels of the upper left quadrant are delivered on tap 1 starting with the upper left pixel and ending with the lower right pixel of the region.

The pixels of the upper right quadrant are delivered on tap 2 starting with the upper rightmost pixel and ending with the lower left pixel of the region.

The pixels of the lower left quadrant are delivered on tap 3 starting with the lower left pixel and ending with the upper right pixel of the region.

The pixels of the lower right quadrant are delivered on tap 4 starting with the lower rightmost pixel and ending with the upper left pixel of the region.

1X4 designates the tap geometry of a four-tap line-scan camera having 1 region across the X-direction.

The pixels are delivered four at a time on four taps beginning with the four leftmost pixels and ending with the four rightmost pixels.

4X designates the tap geometry of a four-tap line-scan camera having 4 regions across the X-direction.

The pixels are delivered four at a time on four taps. Each region delivers its pixels on a single-tap using a common scanning scheme beginning with the leftmost pixel and ending with the rightmost pixel.

6.4. I/O Indices Catalog

I/O indices for input lines Base

Index	ConnectorName	InputPinName	InputStyle
1	IO	IIN1	ISO
2	IO	IIN2	ISO
3	IO	IIN3	ISO
4	IO	IIN4	ISO
5	IO	DIN1	DIFF
6	IO	DIN2	DIFF
7	CAMERA	LVAL	CHANNELLINK
8	CAMERA	FVAL	CHANNELLINK
9	CAMERA	DVAL	CHANNELLINK
10	CAMERA	SPARE	CHANNELLINK
11	CAMERA	CK_PRESENT	CHANNELLINK
23	IO	POWER_5V	POWERSTATE5V
24	IO	POWER_12V	POWERSTATE12V

I/O indices for output lines Base

Index	ConnectorName	OutputPinName	OutputStyle
1	IO	IOUT1	ISO
2	IO	IOUT2	ISO
3	IO	IOUT3	ISO
4	IO	IOUT4	ISO
7	CAMERA	CC1	CHANNELLINK
8	CAMERA	CC2	CHANNELLINK
9	CAMERA	CC3	CHANNELLINK
10	CAMERA	CC4	CHANNELLINK
25	BRACKET	LED	NA

I/O indices for input lines DualBase

Index	ConnectorName	InputPinName	InputStyle
1	IO_A	IIN1	ISO
2	IO_A	IIN2	ISO
3	IO_A	IIN3	ISO
4	IO_A	IIN4	ISO
5	IO_A	DIN1	DIFF
6	IO_A	DIN2	DIFF
7	CAMERA_A	LVAL	CHANNELLINK
8	CAMERA_A	FVAL	CHANNELLINK
9	CAMERA_A	DVAL	CHANNELLINK
10	CAMERA_A	SPARE	CHANNELLINK
11	CAMERA_A	CK_PRESENT	CHANNELLINK
12	IO_B	IIN1	ISO
13	IO_B	IIN2	ISO
14	IO_B	IIN3	ISO
15	IO_B	IIN4	ISO
16	IO_B	DIN1	DIFF
17	IO_B	DIN2	DIFF
18	CAMERA_B	LVAL	CHANNELLINK
19	CAMERA_B	FVAL	CHANNELLINK
20	CAMERA_B	DVAL	CHANNELLINK
21	CAMERA_B	SPARE	CHANNELLINK
22	CAMERA_B	CK_PRESENT	CHANNELLINK
23	IO_A	POWER_5V	POWERSTATE5V
24	IO_A	POWER_12V	POWERSTATE12V
25	IO_B	POWER_5V	POWERSTATE5V
26	IO_B	POWER_12V	POWERSTATE12V

I/O indices for output lines DualBase

Index	ConnectorName	OutputPinName	OutputStyle
1	IO_A	IOUT1	ISO
2	IO_A	IOUT2	ISO
3	IO_A	IOUT3	ISO
4	IO_A	IOUT4	ISO
7	CAMERA_A	CC1	CHANNELLINK
8	CAMERA_A	CC2	CHANNELLINK
9	CAMERA_A	CC3	CHANNELLINK
10	CAMERA_A	CC4	CHANNELLINK
12	IO_B	IOUT1	ISO
13	IO_B	IOUT2	ISO
14	IO_B	IOUT3	ISO
15	IO_B	IOUT4	ISO
18	CAMERA_B	CC1	CHANNELLINK
19	CAMERA_B	CC2	CHANNELLINK
20	CAMERA_B	CC3	CHANNELLINK
21	CAMERA_B	CC4	CHANNELLINK
22	BRACKET	LED_A	NA
28	BRACKET	LED_B	NA

I/O indices for input lines Full FullXR

Index	ConnectorName	InputPinName	InputStyle
1	IO	IIN1	ISO
2	IO	IIN2	ISO
3	IO	IIN3	ISO
4	IO	IIN4	ISO
5	IO	DIN1	DIFF
6	IO	DIN2	DIFF
7	CAMERA	LVAL_X	CHANNELLINK
8	CAMERA	FVAL_X	CHANNELLINK
9	CAMERA	DVAL_X	CHANNELLINK
10	CAMERA	SPARE_X	CHANNELLINK
11	CAMERA	CK_PRESENT_X	CHANNELLINK
12	CAMERA	LVAL_Y	CHANNELLINK
13	CAMERA	FVAL_Y	CHANNELLINK
14	CAMERA	DVAL_Y	CHANNELLINK
15	CAMERA	SPARE_Y	CHANNELLINK
16	CAMERA	CK_PRESENT_Y	CHANNELLINK
17	CAMERA	LVAL_Z	CHANNELLINK
18	CAMERA	FVAL_Z	CHANNELLINK
19	CAMERA	DVAL_Z	CHANNELLINK
20	CAMERA	SPARE_Z	CHANNELLINK
21	CAMERA	CK_PRESENT_Z	CHANNELLINK
23	IO	POWER_5V	POWERSTATE5V
24	IO	POWER_12V	POWERSTATE12V

NOTE
The I/O indices 0 and 22 have no input-related function.

I/O indices for output lines Full FullXR

Index	ConnectorName	OutputPinName	OutputStyle
1	IO	IOUT1	ISO
2	IO	IOUT2	ISO
3	IO	IOUT3	ISO
4	IO	IOUT4	ISO
7	CAMERA	CC1	CHANNELLINK
8	CAMERA	CC2	CHANNELLINK
9	CAMERA	CC3	CHANNELLINK
10	CAMERA	CC4	CHANNELLINK
25	BRACKET	LED	NA

NOTE
The I/O indices 0, 5, 6, and {11 24} have no output-related function.

6.5. Automatic Switching

Refer to the "Automatic Switching" on page 495 section in *D402EN-MultiCam User Guide* PDF document.

6.6. Board Security Feature

A security feature is incorporated in all MultiCam-compliant boards.

The general idea is that the OEM application programmer is able to engrave in the board a secret proprietary key.

The security key is an 8-bytes string of ASCII characters. Any character is allowed. A null character acts as the termination character of the key.

The security key is stored in the non-volatile memory of the board and cannot be read back.

There is no way to obtain this security key number back from the board. However, it is possible to verify that a given board currently holds a security key equal to a given one.

Using this simple mechanism, it is easy to lock an application to a board or to a set of boards.

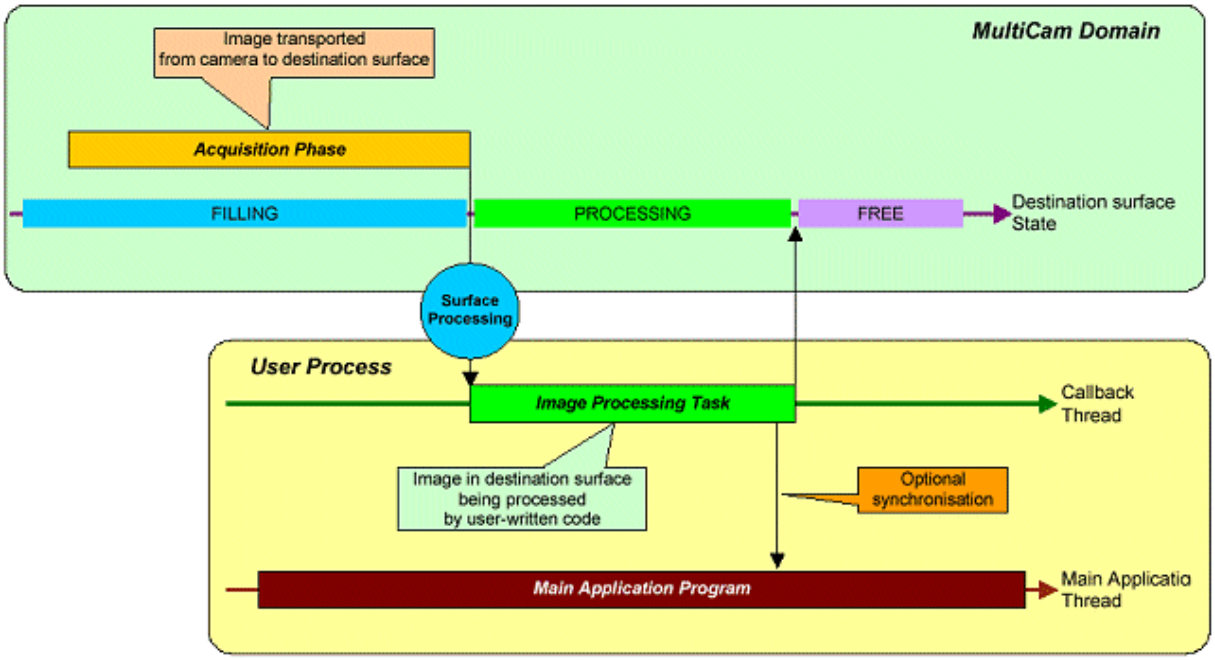
6.7. Callback Signaling

Callback Signaling Mechanism

The callback mechanism implies an event driven behavior. The following description uses the Surface Processing signal as an example of callback generating event.

The Surface Processing signal occurs when a transfer phase terminates. It is issued by a channel to indicate that the destination memory surface has been filled with an image coming from the source camera, and that this surface is available for image processing (see SurfaceState).

The image processing task is performed on this event by a special function called the callback function.



Callback mechanism

The callback function is called by the MultiCam driver, not by the user application. This ensures that the image-processing task is realized at the ideal instant, exactly when the surface becomes ready for processing.

MultiCam benefits from several built-in features to ease the implementation of the callback function.

- A dedicated thread is created for the callback function execution.
- The callback function prototype is declared in the MultiCam system C header file.
- Means are provided to designate the channel and the signal(s) issuing the callback function calls.
- The callback function argument provides all relevant information to the user-written code.

The MultiCam function to register a callback function to a channel is McRegisterCallback.

Callback Signaling Information

Callback Function Prototype

The callback function prototype is declared in the MultiCam system's MultiCam.h header file as follows:

```
typedef void (MCAPI *PMCCALLBACK)(PMCSIGNALINFO SignalInfo);
```

Item	Type	Description
Function	PMCCALLBACK	Callback function
SignalInfo	PMCSIGNALINFO	Argument providing the signal information structure.

The user should define the callback function in the application code in accordance with this prototype.

The callback function is called by the MultiCam driver when a channel issues a pre-defined signal.

The pre-defined signal should be enabled with the SignalEnable parameter. It is allowed to enable several signals.

If more than one enabled signals are issued simultaneously from an object, the callback function is successively called for each signal occurrence.

When the signal occurs, the callback dedicated thread is released, and the callback function is automatically invoked. The thread is restored to an idle condition when the callback function is exited.

The function has a single argument, which is a structure passing information on the signal that caused the callback function. This structure has the *signal information* type.

If the callback signaling mechanism is used, the waiting and advanced signaling mechanisms cannot be used.

Registration of Callback Function

A callback function should be registered to a channel object before use. Only one callback function per object is supported.

Registering the callback function results into the creation in the application process of a separate thread dedicated to the callback function. This thread is maintained in a idle state until a signal occurs. There can be only one dedicated thread per channel object.

A dedicated MultiCam function is provided -for callback registration: McRegisterCallback.

Context

Context is an argument of the callback registration function as well as a member of the signal information structure available to the callback function.

The user is free to use this item at the registration time to hold any identifying information he may find useful.

When the callback function is executed, the user gets back the context information as it was passed to the registration function.

Code Example of Callback Mechanism

The following code uses the callback mechanism to process images grabbed during an acquisition sequence. One or several surfaces have to be created and assigned to the cluster owned by the channel. At the end of each acquisition phase, the surface is filled and made available to the callback function. The Status variable can be used for error checking.

[C]

```
void MyApplication()
{
//Application level initializing code
MCSTATUS Status = McOpenDriver(NULL);
//Application level initializing code
MCHANDLE MyChannel;
Status = McCreateNm("CHANNEL", &MyChannel);
Status = McSetParamInt(MyChannel, MC_DriverIndex, 0);
Status = McSetParamInt(MyChannel, MC_Connector, MC_Connector_M);
//Assign grabber and camera to channel
//Configure channel including triggering mode
//Assign to channel a destination cluster of surfaces
//Registering the callback function
Status = McRegisterCallback(MyChannel, MyFunction, NULL);
//Activating acquisition sequence
Status = McSetParamInt(MyChannel, MC_ChannelState, MC_ChannelState_ACTIVE);
//Acquisition sequence is now active
//A callback is automatically generated after each acquisition phase
//Deleting the channels
Status = McDelete(MyChannel);
//Disconnecting from driver
Status = McCloseDriver();
}
void MCAPI MyFunction(PMCSIGNALINFO SignalInfo){
//...
//Image processing code
//Image to be processed is available in the destination cluster of surfaces
//...
}
```

6.8. Camera Data Transfer Method

The DataLink parameter declares the data transfer method of the camera feeding the channel. MultiCam supports three data transfer methods:

- COMPOSITE: The "composite video" cameras deliver the video data as an analog composite video signal. The signal can be:
  - CVBS including Color, Video, Blanking, and Sync
  - VBS including Video, Blanking, and Sync
- ANALOG: The "analog industrial" cameras deliver the video data as an analog video signal. The signal can be:
  - Single lane VBS including Video, Blanking, and Sync
  - Single lane VB including Video, Blanking
  - Three lane analog RGB with Sync on Green
- CAMERALINK: The "Camera Link" cameras deliver digital video data complying with the Camera Link standard.

NOTE

There is a 1-to-1 match between the values of DataLink and the Euresys frame grabber series: COMPOSITE for Picolo series, ANALOG for Domino series and CAMERALINK for Grablink series.

6.9. Camera Imaging Basic Geometry

The Imaging parameter declares the basic geometry of the camera feeding the channel. MultiCam supports three basic geometries:

- AREA: The area-scan cameras are based on 2D imager(s) and deliver 2D data frames
- LINE: The non-TDI line-scan cameras are based on 1D imager(s) and deliver 1D data lines
- TDI: The TDI line-scan cameras are based on 2D imager(s) and deliver 1D data lines

TDI stands for Time Delay Integration. TDI line-scan cameras exhibit an increased sensitivity since the light integration spans over multiple line periods.

MultiCam distinguishes TDI and non-TDI line-scan cameras since TDI line-scan cameras have specific requirements for their control. However, both are line-scan cameras and share a common set of acquisition modes.

6.10. Camera Spectral Sensitivity

The Spectrum parameter declares the spectral sensitivity of the camera feeding the channel. MultiCam supports three spectral sensitivities:

- BW: The black/white cameras are delivering a monochrome video signal built from an imager having a spectral response covering the visible light spectrum
- IR: The infrared cameras are delivering a monochrome video signal built from an imager having a spectral response covering the infra-red light spectrum
- COLOR: The color cameras are delivering a multi-component video signal built from either a single imager having Color Filter Arrays or from multiple imagers having different spectral responses

For the frame grabber point of view, BW and IR are equivalent. The wording "monochrome cameras" designates both classes of cameras.

The class of color cameras is further divided into several sub-classes. See Color Camera Specification.

6.11. Color Camera Specification

Camera Color Analysis Method

The ColorMethod parameter declares the color analysis method of the camera feeding the channel. MultiCam supports the following color analysis methods:

- NONE: The "monochrome" cameras have no color analysis method.
- RGB: The "RGB" cameras deliver the video data as three separate color components respectively named Red, Green, Blue.
- BAYER: The "Bayer CFA" cameras deliver the raw video obtained from a Bayer CFA imager.
- PRISM: The "PRISM " cameras are a sub-class of RGB cameras using a 3-CCD prism assembly ensuring a perfect registration of all color components of a pixel.
- TRILINEAR: The "trilinear" cameras are a sub-class of non-TDI line-scan RGB color cameras using a triple line-array imager and delivering un-registered color components.

Camera Color Pattern Filter Alignment

The ColorRegistration parameter declares the alignment of the color pattern filter of the camera feeding the channel.

MultiCam supports the following filter alignments for **Bayer CFA cameras**: GB, BG, RG, GR.

MultiCam supports the following filter alignments for **trilinear cameras**: RGB, GBR, BRG.

Color Gap

The ColorGap parameter declares the gap between adjacent sensing lines of the trilinear camera feeding the channel.

This gap is expressed as a number of pixel pitches along the line. It is an unchangeable geometrical feature of the trilinear sensor.

6.12. Channel Creation

To create a channel, go through the following three steps.

1. Create a channel instance.
2. Associate the channel to a board.
3. Select the connector.

Channel Instance Creation

The channel is created with the McCreate or McCreateNm function.

The By-Ident Method

```
McCreate(MC_CHANNEL, &m_Channel);
```

The By-Name Method

```
McCreateNm("CHANNEL", &m_Channel);
```

Maximum number of Channels

- At any time, up to 2048 MultiCam channels can exist in a single process.
- At any time, up to 64 MultiCam channels can exist on a Domino or Grablink board.
- At any time, up to 256 MultiCam channels can exist on a Picolo board.

Channel-Board Association

The targeted board is identified by one of the 4 channel parameters: DriverIndex , PciPosition, BoardName or BoardIdentifier.

Example

```
McSetParamInt(m_Channel, MC_DriverIndex, 0);
```

Connector Selection

After associating the channel with a board, it is required to set the Connector channel parameter.

NOTE

For boards having multiple topologies, it is required to define the BoardTopology before the first channel creation on this board.

Example

```
McSetParamInt(m_Channel, MC_Connector, MC_Connector_VID1);
```

6.13. Code Example: How to Gather Board Information?

The following code scans all installed MultiCam-compliant boards, and builds a database containing their information relative to name, serial number and type.

MC_CONFIGURATION is the C identifier used as a handle to the configuration object. This object has not to be explicitly instantiated.

MC_BOARD is the C identifier used as a handle to the board object. This object has not to be explicitly instantiated.

The Status variable can be used for error checking.

[C]

```
//Defining the database structure type
typedef struct
{  char BoardName[17];
  INT32 SerialNumber;
  INT32 BoardType;
} MULTICAM_BOARDINFO;

//Variables declaration
MULTICAM_BOARDINFO BoardInfo[10];
INT32 BoardCount;
INT32 i;
MCSTATUS Status;

//Connecting to driver
Status = McOpenDriver(NULL);

//Getting number of boards
Status = McGetParamInt(MC_CONFIGURATION, MC_BoardCount, &BoardCount);

//Scanning across MultiCam boards
for (i=0; i<BoardCount; i++)
{

  //Fetching the board name (String MultiCam parameter)
  Status = McGetParamStr(
    MC_BOARD+i,
    MC_BoardName,
    BoardInfo[i].BoardName,
    17);

  //Fetching the board serial number (Integer MultiCam parameter)

   Status = McGetParamInt(
    MC_BOARD+i,
    MC_SerialNumber,
    &BoardInfo[i].SerialNumber);

  //Fetching the board type (Enumerated MultiCam parameter)
  Status = McGetParamInt(
    MC_BOARD+i,
    MC_BoardType,
    &BoardInfo[i].BoardType);
}

//Disconnecting from driver
Status = McCloseDriver();
```

6.14. Enabling Signals

To designate one or several signals as responsible for signaling operation, the MultiCam system provides an adjust-level parameter called SignalEnable.

One such parameter exists for the channel class. It has the MultiCam type "enumerated collection".

Each item of the collection allows for enabling or disabling a specific signal. The value of the item is ON or OFF.

The set of all ON signals constitute the selection of signals enabling the relevant channel to perform one of the following:

- Calling a callback function
- Releasing a waiting thread
- Causing a Windows event

To address a specific signal, the by-ident parameter access method is used with the SignalEnable parameter belonging to the desired channel object. The parameter setting function McSetParamInt or McSetParamStr is used with a parameter identifier established as follows:

To reach signal...	Use parameter identifier...
Frame Trigger Violation	MC_SignalEnable + MC_SIG_FRAME_TRIGGER_VIOLATION
Start Exposure	MC_SignalEnable + MC_SIG_START_EXPOSURE
End Exposure	MC_SignalEnable + MC_SIG_END_EXPOSURE
Release (*)	MC_SignalEnable + MC_SIG_RELEASE
Surface Filled	MC_SignalEnable + MC_SIG_SURFACE_FILLED
Surface Processing	MC_SignalEnable + MC_SIG_SURFACE_PROCESSING
Cluster Unavailable	MC_SignalEnable + MC_SIG_CLUSTER_UNAVAILABLE
Acquisition failure	MC_SignalEnable + MC_SIG_ACQUISITION_FAILURE
End of acquisition	MC_SignalEnable + MC_SIG_END_ACQUISITION_SEQUENCE
Start of acquisition	MC_SignalEnable + MC_SIG_START_ACQUISITION_SEQUENCE
End of channel activity	MC_SignalEnable + MC_SIG_END_CHANNEL_ACTIVITY

(*) This signal is generated only with Domino boards.

Example

The following code enables the "Surface Filled" signal with the channel designated by my_Channel:

```
Status = McSetParamInt (
  my_Channel,
  MC_SignalEnable + MC_SIG_SURFACE_FILLED,
  MC_SignalEnable_ON
);
```

The Status variable can be used for error checking.

6.15. MultiCam Error Codes

Error codes returned by MultiCam functions

Return value	Error identifier	Description
0	MC_OK	No Error
-1	MC_NO_BOARD_FOUND	No Board Found
-2	MC_BAD_PARAMETER	Bad Parameter
-3	MC_IO_ERROR	I/O Error
-4	MC_INTERNAL_ERROR	Internal Error
-5	MC_NO_MORE_RESOURCES	No More Resources
-6	MC_IN_USE	Object still in use
-7	MC_NOT_SUPPORTED	Operation not supported
-8	MC_DATABASE_ERROR	Parameter database error
-9	MC_OUT_OF_BOUND	Value out of bound
-10	MC_INSTANCE_NOT_FOUND	Object instance not found
-11	MC_INVALID_HANDLE	Invalid Handle
-12	MC_TIMEOUT	Timeout
-13	MC_INVALID_VALUE	Invalid Value
-14	MC_RANGE_ERROR	Value not in range
-15	MC_BAD_HW_CONFIG	Invalid hardware configuration
-16	MC_NO_EVENT	No Event
-17	MC_LICENSE_NOT_GRANTED	License not granted
-18	MC_FATAL_ERROR	Fatal error
-19	MC_HW_EVENT_CONFLICT	Hardware event conflict
-20	MC_FILE_NOT_FOUND	File not found
-21	MC_OVERFLOW	Overflow
-22	MC_INVALID_PARAMETER_SETTING	Parameter inconsistency
-23	MC_PARAMETER_ILLEGAL_ACCESS	Illegal operation
-24	MC_CLUSTER_BUSY	Cluster busy
-25	MC_SERVICE_ERROR	MultiCam service error
-26	MC_INVALID_SURFACE	Invalid surface

6.16. Line Rate Modes

Line Rate Mode expresses how the *Downweb Line Rate* is determined in a line-scan acquisition system.

The user specifies the *Line Rate Mode* by means of MultiCam parameter LineRateMode. Five *Line Rate Modes* are identified in MultiCam:

LineRateMode	Description
CAMERA	**Camera** – The *Downweb Line Rate* is originated by the camera.
PULSE	**Trigger Pulse** – The *Downweb Line Rate* originates from a train of pulses applied on the line trigger input belonging to the grabber.
CONVERT	**Rate Converter** – The *Downweb Line Rate* originates from a train of pulses applied on the line trigger input and processed by a rate converter belonging to the grabber.
PERIOD	**Periodic** – The *Downweb Line Rate* originates from an internal periodic generator belonging to the grabber
EXPOSE	**Exposure Time** – The *Downweb Line Rate* is identical to the camera line rate and established by the exposure time settings

LineRateMode = CAMERA

This mode is applicable exclusively for free-run permanent exposure – LxxxxSP – class of line scan cameras when LineCaptureMode = ALL. The grabber does not perform any sampling in the downweb direction; the *Downweb Line Rate* is equal to the camera line rate. The camera line rate is entirely under control of the camera. Notice that most of the line scan cameras provide an internal line rate adjustment.

LineRateMode = PULSE

When the speed of motion is varying, the *Downweb Line Rate* should be slaved to this motion. To achieve this, a motion encoder is a good solution.
The motion encoder delivers an electrical pulse each time the moving web advances by a determined amount of length. The continuous motion results in a train of pulses the frequency of which is proportional to the web speed.
There exists another way to take knowledge of the web speed. In some applications, the motion is caused by a stepping motor controlled by pulses. The controlling train of pulses is also a measure of relative motion.
In both cases, the pulses are called line trigger pulses, and their repetition rate is the Line Trigger Rate. The line trigger pulses are applied to the frame grabber to determine the *Downweb Line Rate* .
Each line trigger pulse may result into the generation of one line in the acquired image. This means that the *Downweb Line Rate* is equal to the Trigger Rate.

LineRateMode = CONVERT

Alternatively to the "PULSE" mode, for more flexibility, the Line Trigger Rate may be scaled up or down to match the required *Downweb Line Rate*. The proportion between the two rates is freely programmable to any value lower or greater than unity, with high accuracy. This makes possible to accommodate a variety of mechanical setups, and still maintain a full control over the downweb resolution. The hardware device responsible for this rate conversion is called the rate converter. This device is a unique characteristic of Euresys line-scan frame grabbers.

LineRateMode = PERIOD

Other circumstances necessitate the *Downweb Line Rate* to be hardware-generated by a programmable timer, called the "periodic generator".

LineRateMode = EXPOSE

Applies to: Base DualBase Full FullXR

This mode is applicable exclusively for line rate controlled permanent exposure – LxxxxRP – class of line scan cameras when LineCaptureMode = ALL. The grabber does not perform any sampling in the downweb direction; the *Downweb Line Rate* is equal to the camera line rate. The camera line rate is entirely under control of the grabber through the exposure time settings.

6.17. MultiCam Storage Formats

Refer to *D406EN MultiCam Storage Formats* PDF document

6.18. MultiCam Tap Geometries

Refer to:

- "TapGeometry Glossary" on page 486 and Supported Tap Geometries topics in *D411EN Grablink Functional Guide* PDF document.
- "TapGeometry" on page 117 topic in *Grablink Parameters Reference* PDF document.

6.19. Using Look-Up Tables

6.20. CAM Files

What Is a CamFile?

The CamFile can be seen as a script of MultiCam setting functions that are played when the CamFile parameter is written to. After the CamFile is played, the channel is ready to operate according to the parameter settings specified in the file. Generally speaking, it means that the channel is ready to start an acquisition for a specified camera in a specified fundamental mode.

"Cam" stands for Camera. In the computer file system, the CamFile exhibits the .cam extension.

A CamFile is a readable ASCII file having the following structure:

- An "CamFile Identification Header" on page 515 (optional)
- A pair of "CamFile Parameter Assignments " on page 516 for the Camera and CamConfig parameters (mandatory)
- A list of "CamFile Parameter Assignments " on page 516 for all relevant MultiCam Channel parameters (optional)

WARNING
A CamFile exclusively contains Channel parameters!

CamFile Identification Header

The identification header is an optional section that includes *MultiCam Studio directives*.

Example of a CamFile header

```
;*************************************************************
; Camera Manufacturer: My Cameras
; Camera Model: ProgressiveFR
; Camera Configuration: Progressive Free-Run Scanning, Analog synchronization
;*************************************************************
```

The MultiCam Studio CamFile directives have the simple format:

```
; <DirectiveName>: <DirectiveValue> <EOL>
```

All values are string of characters terminated by an end of line.

Directive name	Value meaning
Board	Restricts the visibility of the camera in the camera selection wizard of MultiCam Studio. When value is Domino, the CamFile is listed only when the channel is created on a a Domino board.When value is Grablink, the CamFile is listed only when the channel is created on a a Grablink board. Other values are simply ignored. If more than one board directive is present, only the first one is considered
Camera Manufacturer	Declare the manufacturer name to display in the camera selection wizard of MultiCam Studio
Camera Model	Declare the camera model name to display in the camera selection wizard of MultiCam Studio
Revision	Declare the revision number and/or date of the CamFile

CamFile Parameter Assignments

A parameter assignment line has the following format:

```
<ParameterName> = <ParameterValue> [;<Comment>] <EOL>
```

where:

1. ParameterName is a valid MultiCam Channel parameter name for the targeted board.
2. ParameterValue is a valid value for the MultiCam parameter.
3. An optional comment can be appended to the assignment; it must be preceded by a semi-column.
4. A valid End-Of-Line: a CR or a pair of CR and LF characters.

Example of parameters assignment lines

```
Camera = ProgressiveFR
CamConfig = PxxSA      ;
Gain=1000
TargetFrameRate_Hz = 0.5; 1 frame every two seconds
```

Example of comment lines

```
; Camera Specification category
;------------------------------
; Gain=1000
```

NOTE

Only one parameter assignment per line is allowed.

Every line containing a parameter assignment must be terminated by

Spaces or Tab characters can be freely inserted anywhere.

Empty lines, lines containing only comments, are allowed.

WARNING

Considering built-in dependencies between MultiCam parameters, it is recommended to assign values to Channel parameters starting from the parent. Practical rules for Cam Files:

- Keep the statements order of CamFile templates.
- When a parameter statement is added in a CamFile, follow the same order as in the Channel Class section of the Parameters Reference manual.

Loading the CamFile

The loading of a CamFile into a MultiCam channel is a matter of setting the CamFile parameter of a MultiCam channel to the value of the CamFile name (without the .cam extension)

When a CamFile is loaded, it is simply interpreted by the MultiCam driver as a series of "set parameter" function calls.

Examples

The following lines of code implement possible CamFile parameter assignment to a MultiCam channel defined in a Domino board (depends of the camera).

```
MCSTATUS Status = McSetParamStr(MyChannel, MC_CamFile, "VCC-870A_P15RA");
MCSTATUS Status = McSetParamStr(MyChannel, MC_CamFile, "KP-F3_I60SM");
MCSTATUS Status = McSetParamStr(MyChannel, MC_CamFile, "XC-ES30CE_I50SM_R");
```

The following lines of code implement possible camera assignment to a MultiCam channel defined in a Grablink board (depends of the camera).

```
MCSTATUS Status = McSetParamStr(MyChannel, MC_CamFile, "4000m_P16RG");
MCSTATUS Status = McSetParamStr(MyChannel, MC_CamFile, "Colibri2048CL_L2048RG");
```

CamFile libraries

CamFile Templates

A CamFile template is a Camfile intended to be customized by the MultiCam user willing to interface a particular camera with a Domino or a Grablink board.

The MultiCam driver is delivered with a collection of templates. The MultiCam driver installation tool installs the CamFile templates as follows:

The CamFile templates applicable to the the Grablink boards are stored in the <InstallDir>\Cameras_TEMPLATES\Grablink\ directory.

Refer to Interfacing Camera Link Cameras for additional information about CamFile templates for Grablink boards.

Camera Interface Packages Library

A Camera Interface Package is a set of files that contains all the information needed by a MultiCam user to configure a MultiCam channel for a particular camera model. A Camera Interface Package is a ZIP file that includes:

- Ready-to-use CamFiles with the exhaustive set of relevant parameters. One for each of the recommended operating modes
- A documentation explaining how to use this particular camera model with Euresys frame grabbers

When unzipped on the target machine, the CamFiles and the documentation are extracted in the <InstallDir>\Cameras\<Manufacturer>\ folder.

The library of Camera Interface Packages contains a large amount of packages for both analog and Camera Link digital camera models. Furthermore, this library is regularly updated with new packages and constantly growing.

There are 2 ways to access the library:

1. Automatic update with MultiCam Studio

MultiCam Studio provides a convenient way to download and update all the available CamFiles. MultiCam Studio automatically downloads and installs on the MultiCam install directory, from the website, a ZIP file containing the CamFiles and the associated PDF documentation files.

2. Free downloads from the Euresys website

The library directory is available online on https://www.euresys.com/Support/Supported-cameras. The directory can be easily browsed using interactive filters. Each entry in the directory provides the following fields:

- Camera manufacturer name
- Camera model name
- Compatible Euresys boards
- Link to the Camera Interface Package ZIP file