

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

Namespace List

Here is a list of all namespaces with brief descriptions:

ATEsystem_PIRIS

Generated by [doxygen](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

[Classes](#) | [Typedefs](#) | [Enumerations](#) | [Functions](#) |

[Variables](#)

ATEsystem_PIRIS Namespace Reference

Classes

class **CPirisEthernet**

class **CPirisMain**

class **CPirisSerial**

class **DataID**

class **DataParams**

class **DataPosition**

class **DataState**

class **ErrorCluster**

class **Factory**

class **FocusZoomIris**

class **PirisDevice**

class **SmartPointer**

class **StatusEx**

class **Utils**

class **Version**

Typedefs

```
typedef int16_t Status_t
```

```
typedef int16_t DevID_t
```

```
typedef SmartPointer< IDDevice > IPiris
```

```
typedef IDevice *(* CreateDeviceFn) (void)
```

Enumerations

```
enum Mode : uint8_t { SERIAL = 0, ETHERNET = 1 }
```

```
enum Status : int16_t {
    SUCCESS = 0, GENERAL_ERROR = -1, OPEN_FAILED =
    -2, CLOSE_FAILED = -3,
    ALREADY_OPEN = -4, DEVICE_IS_BUSY = -5,
    DEVICE_IS_CLOSED = -6, INVALID_DEVICE = -7,
    WRITE_ERROR = -8, READ_ERROR = -9, SCAN_FAILED =
    -10, INTERNAL_ERROR = -11,
    RX_TIMEOUT = -12, DEV_SET_FAILED = -13,
    WRONG_MODE = -14, DEVICE_NOT_EXIST = -15,
    PARSE_ANSWER_FAIL = -16, UNDEFINED_ERROR = -20
}
```

```
enum State : int8_t { UNKNOWN = 0, CLOSED = 1, BUSY = 2,
READY = 3 }
```

```
enum PirisDeviceType : uint8_t { ETHERNET_DEVICE = 0,
ETHERNET_PYLON = 1, ETHERNET_CAMERA = 2,
SERIAL_UART = 3 }
```

```
enum VerboseLevel : uint8_t { NONE = 0, BASIC = 1, FULL = 2 }
```

```
enum YesNoNA : uint8_t { NOT_AVAILABLE = 0, YES = 1, NO = 2
}
```

Functions

```
std::ostream & operator<< (std::ostream& os, const Status stat)
```

```
std::ostream & operator<< (std::ostream& os, const PirisDeviceType ty)
```

```
virtual Status Open (const PirisDeviceType ty, VerboseLevel verbose=VerboseLevel::Normal)
```

```
virtual Status Close ()=0
```

```
virtual std::tuple<StatusEx, DataID> ReadID ()=0
```

```
virtual std::tuple<StatusEx, DataPosition> ReadPosition ()=0
```

```
virtual std::tuple<StatusEx, DataParams> ReadParams ()=0
```

```
virtual std::tuple<StatusEx, DataState> ReadState ()=0
```

```
virtual StatusEx DevReset ()=0
```

```
virtual StatusEx DevHoming ()=0
```

```
virtual StatusEx SetAbsolute (uint16_t focus, uint16_t zoom=0, uint16_t iris, bool ir_filter=false)=0
```

```
virtual StatusEx SetAbsolute (const FocusZoomIris< uint16_t > &values, bool ir_filter=false)=0
```

```
virtual StatusEx SetRelative (int16_t focus, int16_t zoom=0, int16_t iris)=0
```

```
virtual StatusEx SetRelative (const  
FocusZoomIris< int16_t  
&values)=0
```

```
virtual YesNoNA GetFwPollSupport ()=0
```

```
virtual VerboseLevel GetVerboseLevel ()=0
```

```
virtual void SetVerboseLevel (Verbo  
level)=0
```

```
virtual ~IDevice ()=0
```

```
virtual Status close (bool ignore_err=fa
```

```
virtual Status write (const std::string da
```

```
virtual Status read (std::string &data)=0
```

```
virtual Status flush (bool ignore_err=fa
```

```
virtual void remove ()=0
```

```
virtual ~IComm ()=0
```

Variables

Interface **IDevice**

Interface **IComm**

Typedef Documentation

◆ CreateDeviceFn

```
typedef IDevice*(* ATESystem_PIRIS::CreateDeviceFn) (void)
```

Factory helper function pointer.

◆ DevID_t

```
typedef int16_t ATEsystem_PIRIS::DevID_t
```

Unique device ID, that can be any device manipulated with.

◆ IPiris

```
typedef SmartPointer<IDevice> ATEsystem_PIRIS::IPiris
```

IPiris Smart Pointer is main object, that represents single PIRIS device. No need to delete after.

◆ Status_t

```
typedef int16_t ATEsystem_PIRIS::Status_t
```

Numerical representation of status. 0 is OK, everything else means error.

Enumeration Type Documentation

◆ Mode

```
enum ATEsystem_PIRIS::Mode : uint8_t
```

Main operation mode, which depends on PIRIS device product version. Either RS232 (SERIAL) or ETHERNET.

Enumerator	
SERIAL	
ETHERNET	

◆ PirisDeviceType

```
enum ATEsystem_PIRIS::PirisDeviceType : uint8_t
```

More specific device type. ETHERNET_DEVICE is standalone example case, ETHERNET_PYTHON and ETHERNET_CAMERA means pylon cam instance opened elsewhere and passed to driver.

Enumerator	
ETHERNET_DEVICE	
ETHERNET_PYTHON	
ETHERNET_CAMERA	
SERIAL_UART	

◆ State

```
enum ATEsystem_PIRIS::State : int8_t
```

State machine status. Internal enum used to handle driver states.

Enumerator		
UNKNOWN		
CLOSED		
BUSY		
READY		

◆ Status

```
enum ATEsystem_PIRIS::Status : int16_t
```

Main status output, any action returns this status. 0 is OK, everything else means error.

Enumerator
SUCCESS
GENERAL_ERROR
OPEN_FAILED
CLOSE_FAILED
ALREADY_OPEN
DEVICE_IS_BUSY
DEVICE_IS_CLOSED
INVALID_DEVICE
WRITE_ERROR
READ_ERROR
SCAN_FAILED
INTERNAL_ERROR
RX_TIMEOUT
DEV_SET_FAILED
WRONG_MODE
DEVICE_NOT_EXIST
PARSE_ANSWER_FAIL
UNDEFINED_ERROR

◆ VerboseLevel

```
enum ATEsystem_PIRIS::VerboseLevel : uint8_t
```

Level of verbosity. FULL is used to debug, NONE is for production.

Enumerator	
NONE	
BASIC	
FULL	

◆ YesNoNA

```
enum ATEsystem_PIRIS::YesNoNA : uint8_t
```

Simple 3 state logic handler.

Enumerator		
NOT_AVAILABLE		
YES		
NO		

Function Documentation

◆ Close()

virtual Status ATEsystem_PIRIS::Close()

pure virtual

Close connection with PIRIS device.

Returns

Status class

◆ close()

virtual **Status**

ATEsystem_PIRIS::close (bool **ignore_err = false**) pure virtual

Close connection to PIRIS device. Must be overridden in child.

Parameters

ignore_err If true, force quit connection and discard any exception

Returns

Status class

◆ DevHoming()

virtual **StatusEx** ATEsystem_PIRIS::DevHoming()

pure virtual

Perform a reset of lens of PIRIS device. It takes about 5-15 seconds and time depends on actual motor positions.

Returns

extended Status class **StatusEx**

◆ DevReset()

virtual **StatusEx** ATEsystem_PIRIS::DevReset()

pure virtual

Perform a reset of whole PIRIS device. It takes about 5-15 seconds and time depends on actual motor positions.

Returns

extended Status class **StatusEx**

◆ flush()

virtual **Status**

ATEsystem_PIRIS::flush (bool **ignore_err = false**) pure virtual

Flush comm rx/tx buffer of PIRIS device. Must be overridden in child.

Parameters

ignore_err If true, discard any exception

Returns

Status class

◆ GetFwPollSupport()

virtual **YesNoNA**

ATEsystem_PIRIS::GetFwPollSupport

() pure virtual

Return information about device FW, if it supports continuous poll
(>=1.7.2) returns Yes. More in PDF manual

Returns

3 state logic enum can be Yes, No or NA

◆ GetVerboseLevel()

virtual **VerboseLevel**

ATEsystem_PIRIS::GetVerboseLevel

() pure virtual

Return level of verbosity of current PIRIS instance.

Returns

Level of verbosity can be FULL, BASIC or NONE

◆ Open()

```
virtual Status  
ATEsystem_PIRIS::Open ( const PirisDevice & dev,  
                         VerboseLevel      verbose = VerboseLeve  
                         )
```

Open connection to PIRIS device.

Parameters

dev PirisDevice class

Parameters

verbose level of verbosity

Returns

Status class

◆ operator<<() [1/2]

```
std::ostream & ATESystem_PIRIS::operator<< ( std::ostream & os,
                                                const Status stat
                                              )
```

◆ operator<<() [2/2]

```
std::ostream &
ATEsystem_PIRIS::operator<< ( std::ostream & os,
                                const PirisDeviceType type
                                )
```

◆ **read()**

virtual **Status**

ATEsystem_PIRIS::read

(std::string & **data**)

pure virtual

Read data to PIRIS device. Must be overridden in child.

Parameters

data ASCII data defined by ATEsystem PIRIS protocol
(datasheet)

Returns

Status class

◆ ReadID()

```
virtual std::tuple<StatusEx, DataID>
ATEsystem_PIRIS::ReadID() pure virtual
```

Read ID (name, FW version) from PIRIS device.

For compatibility reason it is obligated to call this method on every start, to set correct timeouts.

Returns

tuple of extended Status class **StatusEx** and data container, yielding the payload

◆ ReadParams()

```
virtual std::tuple<StatusEx, DataParams>
ATEsystem_PIRIS::ReadParams() pure virtual
```

Read parameters (max values of motors, lens type) from PIRIS device.

Returns

tuple of extended Status class **StatusEx** and data container, yielding the payload

◆ ReadPosition()

```
virtual std::tuple<StatusEx, DataPosition>
ATEsystem_PIRIS::ReadPosition() pure virtual
```

Read actual position of motors from PIRIS device.

Returns

tuple of extended Status class **StatusEx** and data container,
yielding the payload

◆ ReadState()

```
virtual std::tuple<StatusEx, DataState>
ATEsystem_PIRIS::ReadState ( ) pure virtual
```

Read status (state of motors, motors are busy) from PIRIS device.

Returns

tuple of extended Status class **StatusEx** and data container,
yielding the payload

◆ remove()

```
virtual void ATESystem_PIRIS::remove( )
```

pure virtual

Remove PIRIS device. Must be overridden in child.

◆ SetAbsolute() [1/2]

virtual **StatusEx**

```
ATEsystem_PIRIS::SetAbsolute ( uint16_t focus = 0,  
                               uint16_t zoom = 0,  
                               uint16_t iris = 0,  
                               bool     ir_filter = false  
                           )
```

pure virtual

Set focus, zoom, iris and ir filter values in absolute mode (every value is ranged from 0 to its max)

Parameters

focus focus value

Parameters

zoom zoom value

Parameters

iris iris value

Parameters

ir_filter ir filter value

Returns

extended Status class **StatusEx**

◆ SetAbsolute() [2/2]

```
virtual StatusEx  
ATEsystem_PIRIS::SetAbsolute ( const FocusZoomIris< uint16_t > &  
                                bool  
                                )
```

Set focus, zoom, iris and ir by container class in absolute mode (every to its max)

Parameters

focus focus value

Parameters

zoom zoom value

Parameters

iris iris value

Parameters

ir_filter ir filter value

Returns

extended Status class **StatusEx**

◆ SetRelative() [1/2]

```
virtual StatusEx  
ATEsystem_PIRIS::SetRelative ( int16_t focus = 0,  
                                int16_t zoom = 0,  
                                int16_t iris = 0  
                                ) pure virtual
```

Set focus, zoom, iris and ir filter values in relative mode (every next value has its base 0 from previous)

Parameters

focus focus value

Parameters

zoom zoom value

Parameters

iris iris value

Parameters

ir_filter ir filter value

Returns

extended Status class **StatusEx**

◆ SetRelative() [2/2]

virtual **StatusEx**

ATEsystem_PIRIS::SetRelative (const **FocusZoomIris**< int16_t > & **v&**

Set focus, zoom, iris and ir by container class in relative mode (every new value is calculated base from previous)

Parameters

focus focus value

Parameters

zoom zoom value

Parameters

iris iris value

Parameters

ir_filter ir filter value

Returns

extended Status class **StatusEx**

◆ SetVerboseLevel()

```
virtual void  
ATEsystem_PIRIS::SetVerboseLevel ( VerboseLevel level ) pure virtual
```

Set the level of verbosity of current PIRIS instance.

Parameters

level Level of verbosity can be FULL, BASIC or NONE

◆ write()

virtual **Status**

ATEsystem_PIRIS::write

(const std::string **data**)

pure virtual

Write data to PIRIS device. Must be overridden in child.

Parameters

data ASCII data defined by ATEsystem PIRIS protocol
(datasheet)

Returns

Status class

◆ ~IComm()

IComm::~IComm ()

pure virtual

◆ ~IDevice()

IDevice::~IDevice ()

pure virtual

Variable Documentation

◆ IComm

Interface ATESystem_PIRIS::IComm

Initial value:

```
{  
public:  
  
    virtual Status open(const PirisDevice& dev,  
        VerboseLevel verbose = VerboseLevel::NONE) =  
        0
```

Internal interface used to provide polymorphic communication functionality (Serial, Ethernet)

◆ IDevice

Interface ATESystem_PIRIS::IDevice

Initial value:

```
{  
public:  
  
    virtual Status Open(void* dev, PirisDeviceType  
        type, VerboseLevel verbose =  
        VerboseLevel::NONE) = 0
```

Main public interface presenting all available methods, that can be called on each PIRIS object by user.

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

Here is a list of all namespace members with links to the namespace documentation for each member:

- a -

- ALREADY_OPEN : [ATEsystem_PIRIS](#)

- b -

- BASIC : [ATEsystem_PIRIS](#)
- BUSY : [ATEsystem_PIRIS](#)

- c -

- Close() : [ATEsystem_PIRIS](#)
- close() : [ATEsystem_PIRIS](#)
- CLOSE_FAILED : [ATEsystem_PIRIS](#)
- CLOSED : [ATEsystem_PIRIS](#)
- CreateDeviceFn : [ATEsystem_PIRIS](#)

- d -

- DEV_SET_FAILED : [ATEsystem_PIRIS](#)
- DevHoming() : [ATEsystem_PIRIS](#)
- DEVICE_IS_BUSY : [ATEsystem_PIRIS](#)
- DEVICE_IS_CLOSED : [ATEsystem_PIRIS](#)
- DEVICE_NOT_EXIST : [ATEsystem_PIRIS](#)
- DevID_t : [ATEsystem_PIRIS](#)
- DevReset() : [ATEsystem_PIRIS](#)

- e -

- ETHERNET : [ATEsystem_PIRIS](#)

- ETHERNET_CAMERA : **ATEsystem_PIRIS**
- ETHERNET_DEVICE : **ATEsystem_PIRIS**
- ETHERNET_PYTHON : **ATEsystem_PIRIS**

- f -

- flush() : **ATEsystem_PIRIS**
- FULL : **ATEsystem_PIRIS**

- g -

- GENERAL_ERROR : **ATEsystem_PIRIS**
- GetFwPollSupport() : **ATEsystem_PIRIS**
- GetVerboseLevel() : **ATEsystem_PIRIS**

- i -

- IComm : **ATEsystem_PIRIS**
- IDevice : **ATEsystem_PIRIS**
- INTERNAL_ERROR : **ATEsystem_PIRIS**
- INVALID_DEVICE : **ATEsystem_PIRIS**
- IPiris : **ATEsystem_PIRIS**

- m -

- Mode : **ATEsystem_PIRIS**

- n -

- NO : **ATEsystem_PIRIS**
- NONE : **ATEsystem_PIRIS**
- NOT_AVAILABLE : **ATEsystem_PIRIS**

- o -

- Open() : **ATEsystem_PIRIS**
- OPEN_FAILED : **ATEsystem_PIRIS**
- operator<<() : **ATEsystem_PIRIS**

- p -

- PARSE_ANSWER_FAIL : **ATEsystem_PIRIS**
- PirisDeviceType : **ATEsystem_PIRIS**

- r -

- read() : **ATEsystem_PIRIS**
- READ_ERROR : **ATEsystem_PIRIS**
- ReadID() : **ATEsystem_PIRIS**
- ReadParams() : **ATEsystem_PIRIS**
- ReadPosition() : **ATEsystem_PIRIS**
- ReadState() : **ATEsystem_PIRIS**
- READY : **ATEsystem_PIRIS**
- remove() : **ATEsystem_PIRIS**
- RX_TIMEOUT : **ATEsystem_PIRIS**

- s -

- SCAN_FAILED : **ATEsystem_PIRIS**
- SERIAL : **ATEsystem_PIRIS**
- SERIAL_UART : **ATEsystem_PIRIS**
- SetAbsolute() : **ATEsystem_PIRIS**
- SetRelative() : **ATEsystem_PIRIS**
- SetVerboseLevel() : **ATEsystem_PIRIS**
- State : **ATEsystem_PIRIS**
- Status : **ATEsystem_PIRIS**
- Status_t : **ATEsystem_PIRIS**
- SUCCESS : **ATEsystem_PIRIS**

- u -

- UNDEFINED_ERROR : **ATEsystem_PIRIS**
- UNKNOWN : **ATEsystem_PIRIS**

- v -

- VerboseLevel : **ATEsystem_PIRIS**

- W -

- write() : **ATEsystem_PIRIS**
- WRITE_ERROR : **ATEsystem_PIRIS**
- WRONG_MODE : **ATEsystem_PIRIS**

- y -

- YES : **ATEsystem_PIRIS**
- YesNoNA : **ATEsystem_PIRIS**

- ~ -

- ~IComm() : **ATEsystem_PIRIS**
- ~IDevice() : **ATEsystem_PIRIS**

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

- Close() : **ATEsystem_PIRIS**
 - close() : **ATEsystem_PIRIS**
 - DevHoming() : **ATEsystem_PIRIS**
 - DevReset() : **ATEsystem_PIRIS**
 - flush() : **ATEsystem_PIRIS**
 - GetFwPollSupport() : **ATEsystem_PIRIS**
 - GetVerboseLevel() : **ATEsystem_PIRIS**
 - Open() : **ATEsystem_PIRIS**
 - operator<<() : **ATEsystem_PIRIS**
 - read() : **ATEsystem_PIRIS**
 - ReadID() : **ATEsystem_PIRIS**
 - ReadParams() : **ATEsystem_PIRIS**
 - ReadPosition() : **ATEsystem_PIRIS**
 - ReadState() : **ATEsystem_PIRIS**
 - remove() : **ATEsystem_PIRIS**
 - SetAbsolute() : **ATEsystem_PIRIS**
 - SetRelative() : **ATEsystem_PIRIS**
 - SetVerboseLevel() : **ATEsystem_PIRIS**
 - write() : **ATEsystem_PIRIS**
 - ~IComm() : **ATEsystem_PIRIS**
 - ~IDevice() : **ATEsystem_PIRIS**
-

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

- IComm : **ATEsystem_PIRIS**
- IDevice : **ATEsystem_PIRIS**

Generated by **doxygen** 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

- CreateDeviceFn : **ATEsystem_PIRIS**
- DevID_t : **ATEsystem_PIRIS**
- IPiris : **ATEsystem_PIRIS**
- Status_t : **ATEsystem_PIRIS**

Generated by **doxygen** 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

- Mode : `ATEsystem_PIRIS`
 - PirisDeviceType : `ATEsystem_PIRIS`
 - State : `ATEsystem_PIRIS`
 - Status : `ATEsystem_PIRIS`
 - VerboseLevel : `ATEsystem_PIRIS`
 - YesNoNA : `ATEsystem_PIRIS`
-

Generated by [doxygen](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

- a -

- ALREADY_OPEN : **ATEsystem_PIRIS**

- b -

- BASIC : **ATEsystem_PIRIS**
- BUSY : **ATEsystem_PIRIS**

- c -

- CLOSE_FAILED : **ATEsystem_PIRIS**
- CLOSED : **ATEsystem_PIRIS**

- d -

- DEV_SET_FAILED : **ATEsystem_PIRIS**
- DEVICE_IS_BUSY : **ATEsystem_PIRIS**
- DEVICE_IS_CLOSED : **ATEsystem_PIRIS**
- DEVICE_NOT_EXIST : **ATEsystem_PIRIS**

- e -

- ETHERNET : **ATEsystem_PIRIS**
- ETHERNET_CAMERA : **ATEsystem_PIRIS**
- ETHERNET_DEVICE : **ATEsystem_PIRIS**
- ETHERNET_PYTHON : **ATEsystem_PIRIS**

- f -

- FULL : **ATEsystem_PIRIS**

- g -

- GENERAL_ERROR : **ATEsystem_PIRIS**

- i -

- INTERNAL_ERROR : **ATEsystem_PIRIS**
- INVALID_DEVICE : **ATEsystem_PIRIS**

- n -

- NO : **ATEsystem_PIRIS**
- NONE : **ATEsystem_PIRIS**
- NOT_AVAILABLE : **ATEsystem_PIRIS**

- o -

- OPEN_FAILED : **ATEsystem_PIRIS**

- p -

- PARSE_ANSWER_FAIL : **ATEsystem_PIRIS**

- r -

- READ_ERROR : **ATEsystem_PIRIS**
- READY : **ATEsystem_PIRIS**
- RX_TIMEOUT : **ATEsystem_PIRIS**

- s -

- SCAN_FAILED : **ATEsystem_PIRIS**
- SERIAL : **ATEsystem_PIRIS**
- SERIAL_UART : **ATEsystem_PIRIS**
- SUCCESS : **ATEsystem_PIRIS**

- u -

- UNDEFINED_ERROR : **ATEsystem_PIRIS**
- UNKNOWN : **ATEsystem_PIRIS**

- w -

- WRITE_ERROR : **ATEsystem_PIRIS**
- WRONG_MODE : **ATEsystem_PIRIS**

- y -

- YES : **ATEsystem_PIRIS**

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

Class List

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

[detail level [1](#) [2](#)]

▼ [N](#) ATEsystem_PIRIS

- [C CPirisEthernet](#)
- [C CPirisMain](#)
- [C CPirisSerial](#)
- [C DataID](#)
- [C DataParams](#)
- [C DataPosition](#)
- [C DataState](#)
- [C ErrorCluster](#)
- [C Factory](#)
- [C FocusZoomIris](#)
- [C PirisDevice](#)
- [C SmartPointer](#)
- [C StatusEx](#)
- [C Utils](#)
- [C Version](#)

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATESystem_PIRIS > CPirisEthernet

[Public Member Functions](#) |

[Static Public Member Functions](#) |

[Protected Member Functions](#) |

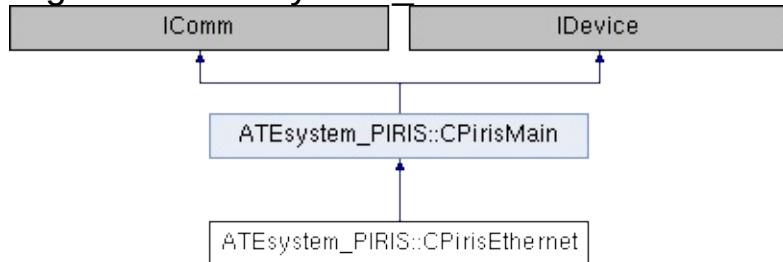
[List of all members](#)

ATESystem_PIRIS::CPirisEthernet Class Reference final

More...

```
#include <ATEsystem.PIRIS_ethernet.h>
```

Inheritance diagram for ATESystem_PIRIS::CPirisEthernet:



Public Member Functions

`CPirisEthernet ()`

`virtual ~CPirisEthernet ()`

`virtual void remove ()`

▶ Public Member Functions inherited from
`ATEsystem_PIRIS::CPirisMain`

`Status Open (void *dev,
PirisDeviceType type,
VerboseLevel
verbose=VerboseLevel::NONE)`

`Status Open (const PirisDevice &dev,
VerboseLevel
verbose=VerboseLevel::NONE)`

`Status Close ()`

`std::tuple< StatusEx, DataID > ReadID ()`

`std::tuple< StatusEx, DataPosition > ReadPosition ()`

`std::tuple< StatusEx, DataParams > ReadParams ()`

`std::tuple< StatusEx, DataState > ReadState ()`

`StatusEx DevReset ()`

`StatusEx DevHoming ()`

`StatusEx SetAbsolute (uint16_t
focus=0, uint16_t zoom=0,
uint16_t iris=0, bool`

```
    ir_filter=false)
```

```
StatusEx SetAbsolute (const
FocusZoomIris< uint16_t >
&values, bool ir_filter=false)
```

```
StatusEx SetRelative (int16_t focus=0,
int16_t zoom=0, int16_t iris=0)
```

```
StatusEx SetRelative (const
FocusZoomIris< int16_t >
&values)
```

```
YesNoNA GetFwPollSupport ()
```

```
VerboseLevel GetVerboseLevel ()
```

```
void SetVerboseLevel
(VerboseLevel level)
```

```
CPirisMain ()
```

```
virtual ~CPirisMain ()
```

Static Public Member Functions

```
static int16_t scan (std::vector< Pylon::CDeviceInfo > &devices,  
                      bool verbose=false)
```

```
static IDevice * create ()
```

▶ Static Public Member Functions inherited from
ATEsystem_PIRIS::CPirisMain

```
static int16_t ScanEthernet (std::vector< Pylon::CDeviceInfo >  
                           &devices, bool verbose=false)
```

```
static int16_t ScanSerial (std::vector< serial::PortInfo > &devices,  
                           bool verbose=false)
```

Protected Member Functions

```
virtual Status open (const PirisDevice &dev, VerboseLevel  
verbose=VerboseLevel::NONE)
```

```
virtual Status close (bool ignore_err=false)
```

```
virtual Status write (const std::string data)
```

```
virtual Status read (std::string &data)
```

```
virtual Status flush (bool ignore_err=false)
```

► Protected Member Functions inherited from
ATEsystem_PIRIS::CPirisMain

```
void state_set (State _state)
```

```
State state_get ()
```

```
bool state_is (State _state)
```

```
void state_reset ()
```

```
bool verbose_is (VerboseLevel _verbose)
```

```
void verbose_set (VerboseLevel _verbose)
```

Additional Inherited Members

- ▶ **Protected Attributes inherited from**

ATEsystem_PIRIS::CPirisMain

```
uint16_t rx_timeout = RX_TIMEOUT_NEW_FW
```

Detailed Description

Main final child class providing Ethernet communication functionality.

Constructor & Destructor Documentation

◆ CPirisEthernet()

```
ATEsystem_PIRIS::CPirisEthernet::CPirisEthernet( )
```

◆ ~CPirisEthernet()

ATEsystem_PIRIS::CPirisEthernet::~CPirisEthernet()

virtual

Member Function Documentation

◆ close()

Status

ATEsystem_PIRIS::CPirisEthernet::close (bool ignore_err = false)

pi

◆ create()

IDevice * ATEsystem_PIRIS::CPirisEthernet::create()

static

◆ flush()

Status

```
ATEsystem_PIRIS::CPirisEthernet::flush ( bool ignore_err = false ) [pri]
```

◆ open()

Status

```
ATEsystem_PIRIS::CPirisEthernet::open ( const PirisDevice & dev,  
                                         VerboseLevel      verbose  
                                         )
```

◆ **read()**

Status

ATEsystem_PIRIS::CPirisEthernet::read (std::string & **data**)

protected

◆ remove()

```
void ATESystem_PIRIS::CPirisEthernet::remove( )
```

virtual

◆ scan()

```
int16_t  
ATEsystem_PIRIS::CPirisEthernet::scan ( std::vector< Pylon::CDeviceInfo>  
                                         bool  
                                         )
```

◆ write()

Status

ATEsystem_PIRIS::CPirisEthernet::write (const std::string & data) protected

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

- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/include/**ATEsystem.PIRIS_ethernet.h**
- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/src/**ATEsystem.PIRIS_ethernet.cpp**

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS

CPirisMain

[Public Member Functions](#) |

[Static Public Member Functions](#) |

[Protected Member Functions](#) |

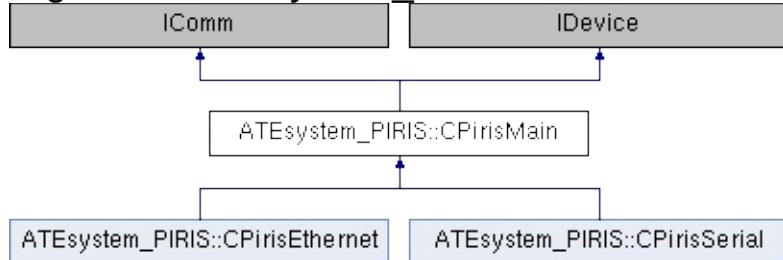
[Protected Attributes](#) | [List of all members](#)

ATEsystem_PIRIS::CPirisMain Class Reference

[More...](#)

```
#include <ATEsystem.PIRIS.h>
```

Inheritance diagram for ATEsystem_PIRIS::CPirisMain:



Public Member Functions

Status **Open** (void *dev,
PirisDeviceType type,
VerboseLevel
verbose=VerboseLevel::NONE)

Status **Open** (const **PirisDevice** &dev,
VerboseLevel
verbose=VerboseLevel::NONE)

Status **Close** ()

std::tuple< **StatusEx**, **DataID** > **ReadID** ()

std::tuple< **StatusEx**, **DataPosition** > **ReadPosition** ()

std::tuple< **StatusEx**, **DataParams** > **ReadParams** ()

std::tuple< **StatusEx**, **DataState** > **ReadState** ()

StatusEx **DevReset** ()

StatusEx **DevHoming** ()

StatusEx **SetAbsolute** (uint16_t
focus=0, uint16_t zoom=0,
uint16_t iris=0, bool
ir_filter=false)

StatusEx **SetAbsolute** (const
FocusZoomIris< uint16_t >
&values, bool ir_filter=false)

StatusEx **SetRelative** (int16_t focus=0,
int16_t zoom=0, int16_t iris=0)

StatusEx **SetRelative** (const
FocusZoomIris< int16_t >
&values)

YesNoNA **GetFwPollSupport** ()

VerboseLevel **GetVerboseLevel** ()

void **SetVerboseLevel**
(**VerboseLevel** level)

CPirisMain ()

virtual ~**CPirisMain** ()

Static Public Member Functions

```
static int16_t ScanEthernet (std::vector< Pylon::CDeviceInfo >  
    &devices, bool verbose=false)
```

```
static int16_t ScanSerial (std::vector< serial::PortInfo > &devices,  
    bool verbose=false)
```

Protected Member Functions

```
void state_set (State _state)
```

```
State state_get ()
```

```
bool state_is (State _state)
```

```
void state_reset ()
```

```
bool verbose_is (VerboseLevel _verbose)
```

```
void verbose_set (VerboseLevel _verbose)
```

Protected Attributes

```
uint16_t rx_timeout = RX_TIMEOUT_NEW_FW
```

Detailed Description

Main abstract class with all key functionality, need to have overridden
comm func in derived class.

Constructor & Destructor Documentation

◆ CPirisMain()

```
ATEsystem_PIRIS::CPirisMain::CPirisMain ( )
```

◆ ~CPirisMain()

```
virtual ATESystem_PIRIS::CPirisMain::~CPirisMain( )
```

inline virtual

Member Function Documentation

◆ Close()

Status ATESystem_PIRIS::CPirisMain::Close()

◆ DevHoming()

StatusEx ATEsystem_PIRIS::CPirisMain::DevHoming()

◆ DevReset()

StatusEx ATEsystem_PIRIS::CPirisMain::DevReset()

◆ GetFwPollSupport()

```
YesNoNA ATESystem_PIRIS::CPirisMain::GetFwPollSupport( )
```

◆ GetVerboseLevel()

VerboseLevel ATESystem_PIRIS::CPirisMain::GetVerboseLevel ()

◆ Open() [1/2]

Status

```
ATEsystem_PIRIS::CPirisMain::Open ( void *           dev,  
                                    PirisDeviceType type,  
                                    VerboseLevel   verbose = VerboseLevel::Info  
                                    )
```

◆ Open() [2/2]

Status

```
ATEsystem_PIRIS::CPirisMain::Open ( const PirisDevice & dev,  
                                     VerboseLevel      verbose =  
                                     )
```

◆ ReadID()

```
std::tuple< StatusEx, DataID >
ATEsystem_PIRIS::CPirisMain::ReadID ( )
```

◆ ReadParams()

```
std::tuple< StatusEx, DataParams >
ATEsystem_PIRIS::CPirisMain::ReadParams ( )
```

◆ ReadPosition()

```
std::tuple< StatusEx, DataPosition >
ATEsystem_PIRIS::CPirisMain::ReadPosition ( )
```

◆ ReadState()

```
std::tuple< StatusEx, DataState >
ATEsystem_PIRIS::CPirisMain::ReadState ( )
```

◆ ScanEthernet()

```
int16_t  
ATEsystem_PIRIS::CPirisMain::ScanEthernet ( std::vector< Pylon::CDe  
                                              bool  
                                              )
```

◆ ScanSerial()

```
int16_t  
ATEsystem_PIRIS::CPirisMain::ScanSerial ( std::vector< serial::PortInfo>  
                                         bool  
                                         )
```

◆ SetAbsolute() [1/2]

StatusEx

```
ATEsystem_PIRIS::CPirisMain::SetAbsolute ( uint16_t focus = 0,  
                                         uint16_t zoom = 0,  
                                         uint16_t iris = 0,  
                                         bool     ir_filter = false  
                                         )
```

◆ SetAbsolute() [2/2]

StatusEx

```
ATEsystem_PIRIS::CPirisMain::SetAbsolute ( const FocusZoomIris< u  
                                         bool  
                                         )
```

◆ SetRelative() [1/2]

StatusEx

```
ATEsystem_PIRIS::CPirisMain::SetRelative ( int16_t focus = 0,  
                                         int16_t zoom = 0,  
                                         int16_t iris = 0  
                                         )
```

◆ SetRelative() [2/2]

StatusEx

ATEsystem_PIRIS::CPirisMain::SetRelative (const **FocusZoomIris**< in

◆ SetVerboseLevel()

```
void  
ATEsystem_PIRIS::CPirisMain::SetVerboseLevel ( VerboseLevel level
```

◆ state_get()

State ATEsystem_PIRIS::CPirisMain::state_get()

protected

◆ state_is()

```
bool  
ATEsystem_PIRIS::CPirisMain::state_is      ( State _state ) protected
```

◆ state_reset()

void ATESystem_PIRIS::CPirisMain::state_reset()

protected

◆ state_set()

```
void  
ATEsystem_PIRIS::CPirisMain::state_set ( State _state ) protected
```

◆ verbose_is()

```
bool  
ATEsystem_PIRIS::CPirisMain::verbose_is ( VerboseLevel _verbose )
```

◆ verbose_set()

```
void  
ATEsystem_PIRIS::CPirisMain::verbose_set ( VerboseLevel _verbose
```

Member Data Documentation

◆ rx_timeout

```
uint16_t ATESystem_PIRIS::CPirisMain::rx_timeout =  
RX_TIMEOUT_NEW_FW
```

protected

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

- ATESystem.PIRIS-driver/ATESystem.PIRIS-driver-Shared/include/[ATESystem.PIRIS.h](#)
- ATESystem.PIRIS-driver/ATESystem.PIRIS-driver-Shared/src/[ATESystem.PIRIS.cpp](#)

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATESystem_PIRIS > CPirisSerial >

[Public Member Functions](#) |

[Static Public Member Functions](#) |

[Protected Member Functions](#) |

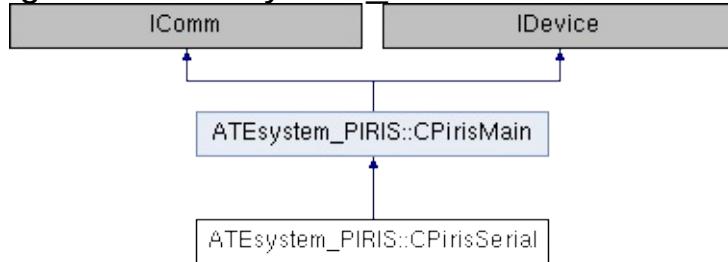
[List of all members](#)

ATEsystem_PIRIS::CPirisSerial Class Reference final

More...

```
#include <ATEsystem.PIRIS_serial.h>
```

Inheritance diagram for ATEsystem_PIRIS::CPirisSerial:



Public Member Functions

`CPirisSerial ()`

`virtual ~CPirisSerial ()`

`virtual void remove ()`

▶ Public Member Functions inherited from
`ATEsystem_PIRIS::CPirisMain`

`Status Open (void *dev,
PirisDeviceType type,
VerboseLevel
verbose=VerboseLevel::NONE)`

`Status Open (const PirisDevice &dev,
VerboseLevel
verbose=VerboseLevel::NONE)`

`Status Close ()`

`std::tuple< StatusEx, DataID > ReadID ()`

`std::tuple< StatusEx, DataPosition > ReadPosition ()`

`std::tuple< StatusEx, DataParams > ReadParams ()`

`std::tuple< StatusEx, DataState > ReadState ()`

`StatusEx DevReset ()`

`StatusEx DevHoming ()`

`StatusEx SetAbsolute (uint16_t
focus=0, uint16_t zoom=0,
uint16_t iris=0, bool`

```
    ir_filter=false)
```

```
StatusEx SetAbsolute (const
FocusZoomIris< uint16_t >
&values, bool ir_filter=false)
```

```
StatusEx SetRelative (int16_t focus=0,
int16_t zoom=0, int16_t iris=0)
```

```
StatusEx SetRelative (const
FocusZoomIris< int16_t >
&values)
```

```
YesNoNA GetFwPollSupport ()
```

```
VerboseLevel GetVerboseLevel ()
```

```
void SetVerboseLevel
(VerboseLevel level)
```

```
CPirisMain ()
```

```
virtual ~CPirisMain ()
```

Static Public Member Functions

```
static int16_t scan (std::vector< serial::PortInfo > &devices, bool  
verbose=false)
```

```
static IDevice * create ()
```

► Static Public Member Functions inherited from
ATEsystem_PIRIS::CPirisMain

```
static int16_t ScanEthernet (std::vector< Pylon::CDeviceInfo >  
&devices, bool verbose=false)
```

```
static int16_t ScanSerial (std::vector< serial::PortInfo > &devices,  
bool verbose=false)
```

Protected Member Functions

```
virtual Status open (const PirisDevice &dev, VerboseLevel  
verbose=VerboseLevel::NONE)
```

```
virtual Status close (bool ignore_err=false)
```

```
virtual Status write (const std::string data)
```

```
virtual Status read (std::string &data)
```

```
virtual Status flush (bool ignore_err=false)
```

► Protected Member Functions inherited from
ATEsystem_PIRIS::CPirisMain

```
void state_set (State _state)
```

```
State state_get ()
```

```
bool state_is (State _state)
```

```
void state_reset ()
```

```
bool verbose_is (VerboseLevel _verbose)
```

```
void verbose_set (VerboseLevel _verbose)
```

Additional Inherited Members

- ▶ **Protected Attributes inherited from**

ATEsystem_PIRIS::CPirisMain

```
uint16_t rx_timeout = RX_TIMEOUT_NEW_FW
```

Detailed Description

Main final child class providing RS232 (Serial) communication functionality.

Constructor & Destructor Documentation

◆ CPirisSerial()

```
ATEsystem_PIRIS::CPirisSerial::CPirisSerial( )
```

◆ ~CPirisSerial()

ATEsystem_PIRIS::CPirisSerial::~CPirisSerial()

virtual

Member Function Documentation

◆ close()

Status

ATEsystem_PIRIS::CPirisSerial::close (bool ignore_err = false) protected

◆ create()

IDevice * ATEsystem_PIRIS::CPirisSerial::create ()

static

◆ flush()

Status

ATEsystem_PIRIS::CPirisSerial::flush (bool ignore_err = false) [protec]

◆ open()

Status

```
ATEsystem_PIRIS::CPirisSerial::open ( const PirisDevice & dev,  
                                      VerboseLevel      verbose =  
                                      )
```

◆ **read()**

Status

ATEsystem_PIRIS::CPirisSerial::read (std::string & **data**) protected virtual

◆ remove()

```
void ATESystem_PIRIS::CPirisSerial::remove( )
```

virtual

◆ scan()

```
int16_t  
ATEsystem_PIRIS::CPirisSerial::scan ( std::vector< serial::PortInfo > &  
                                      bool  
                                      )
```

◆ write()

Status

ATEsystem_PIRIS::CPirisSerial::write (const std::string **data**) protected

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

- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/include/**ATEsystem.PIRIS_serial.h**
 - ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/src/**ATEsystem.PIRIS_serial.cpp**
-

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS

DataID

[Public Member Functions](#) | [Public Attributes](#) |
[List of all members](#)

ATEsystem_PIRIS::DataID Class Reference

[More...](#)

```
#include <ATEsystem.PIRIS.h>
```

Public Member Functions

DataID (std::string **name**="", Version **version**=Version())

Public Attributes

const std::string **name**

const **Version** **version**

Detailed Description

Data container class to provide PIRIS name and version.

Constructor & Destructor Documentation

◆ DataID()

```
ATEsystem_PIRIS::DataID::DataID ( std::string name = "",  
                                  Version    version = Version()  
                                )
```

Member Data Documentation

◆ name

```
const std::string ATEsystem_PIRIS::DataID::name
```

◆ version

```
const Version ATESystem_PIRIS::DataID::version
```

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

- ATESystem.PIRIS-driver/ATESystem.PIRIS-driver-Shared/include/[ATESystem.PIRIS.h](#)
 - ATESystem.PIRIS-driver/ATESystem.PIRIS-driver-Shared/src/[ATESystem.PIRIS.cpp](#)
-

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS

DataParams

[Public Member Functions](#) | [Public Attributes](#) |
[List of all members](#)

ATEsystem_PIRIS::DataParams Class Reference

[More...](#)

```
#include <ATEsystem.PIRIS.h>
```

Public Member Functions

DataParams (**FocusZoomIris< uint16_t > max=****FocusZoomIris< uint16_t >()**, std::string **lens=""**, bool ir=false, bool sens=false)

Public Attributes

const **FocusZoomIris< uint16_t >** **max_value**

const std::string **lens**

const bool **ir_present**

const bool **sens_present**

Detailed Description

Params container class to provide PIRIS motor max values, lens type, ir and sensor present flag.

Constructor & Destructor Documentation

◆ DataParams()

```
ATEsystem_PIRIS::DataParams::DataParams ( FocusZoomIris< uint1
                                         std::string
                                         bool
                                         bool
                                         )
```

Member Data Documentation

◆ ir_present

```
const bool ATESystem_PIRIS::DataParams::ir_present
```

◆ lens

```
const std::string ATEsystem_PIRIS::DataParams::lens
```

◆ max_value

```
const FocusZoomIris<uint16_t>
ATEsystem_PIRIS::DataParams::max_value
```

◆ sens_present

```
const bool ATESystem_PIRIS::DataParams::sens_present
```

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

- ATESystem.PIRIS-driver/ATESystem.PIRIS-driver-Shared/include/[ATESystem.PIRIS.h](#)
- ATESystem.PIRIS-driver/ATESystem.PIRIS-driver-Shared/src/[ATESystem.PIRIS.cpp](#)

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS

DataPosition

[Public Member Functions](#) | [Public Attributes](#) |
[List of all members](#)

ATEsystem_PIRIS::DataPosition Class Reference

[More...](#)

```
#include <ATEsystem.PIRIS.h>
```

Public Member Functions

DataPosition (**FocusZoomIris< uint16_t >** pos=**FocusZoomIris< uint16_t >()**, **bool ir=false**)

Public Attributes

const **FocusZoomIris< uint16_t >** **position**

const bool **ir_filter**

Detailed Description

Position container class to provide PIRIS motor actual positions and ir filter state.

Constructor & Destructor Documentation

◆ DataPosition()

```
ATEsystem_PIRIS::DataPosition::DataPosition ( FocusZoomIris< uint1
                                              bool
                                              )
```

Member Data Documentation

◆ ir_filter

```
const bool ATESystem_PIRIS::DataPosition::ir_filter
```

◆ position

```
const FocusZoomIris<uint16_t>
ATEsystem_PIRIS::DataPosition::position
```

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

- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/include/[ATEsystem.PIRIS.h](#)
- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/src/[ATEsystem.PIRIS.cpp](#)

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS

DataState

[Public Member Functions](#) | [Public Attributes](#) |
[List of all members](#)

ATEsystem_PIRIS::DataState Class Reference

More...

```
#include <ATEsystem.PIRIS.h>
```

Public Member Functions

DataState (FocusZoomIris< bool > state=FocusZoomIris< bool > (), bool busy=false)

Public Attributes

const **FocusZoomIris**< bool > **state**

const bool **motors_busy**

Detailed Description

State container class to provide PIRIS motor status (OK/NOK) and motor busy information

Constructor & Destructor Documentation

◆ DataState()

```
ATEsystem_PIRIS::DataState::DataState ( FocusZoomIris< bool > sta  
                                         bool                         bus  
                                         )
```

Member Data Documentation

◆ motors_busy

```
const bool ATESystem_PIRIS::DataState::motors_busy
```

◆ state

```
const FocusZoomIris<bool> ATEsystem_PIRIS::DataState::state
```

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

- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/include/[ATEsystem.PIRIS.h](#)
 - ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/src/[ATEsystem.PIRIS.cpp](#)
-

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATESystem_PIRIS

ErrorCluster

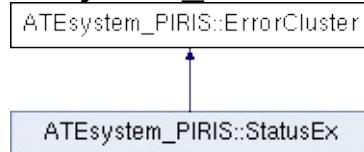
[Public Member Functions](#) | [Friends](#) |
[List of all members](#)

ATEsystem_PIRIS::ErrorCluster Class Reference

[More...](#)

```
#include <ATEsystem.PIRIS.h>
```

Inheritance diagram for ATEsystem_PIRIS::ErrorCluster:



Public Member Functions

ErrorCluster (std::string
err_raw="", bool
override=false)

const bool **getOk** () const

const int8_t **getErrNum** () const

const std::string **getMsg** () const

const std::tuple< bool, int8_t, std::string > **getErrorCluster** () const

std::string **ToString** () const

Friends

std::ostream & **operator<<** (std::ostream &strm, const **ErrorCluster** &err)

Detailed Description

Class with complete error information. That is ok (T = not error), err_num and msg.

Constructor & Destructor Documentation

◆ ErrorCluster()

```
ATEsystem_PIRIS::ErrorCluster::ErrorCluster ( std::string err_raw = "",  
                                              bool      overide = fal  
                                              )
```

Member Function Documentation

◆ getErrNum()

```
const int8_t ATEsystem_PIRIS::ErrorCluster::getErrNum( ) const
```

◆ getErrorCluster()

```
const std::tuple< bool, int8_t, std::string >
ATEsystem_PIRIS::ErrorCluster::getErrorCluster ( ) const
```

◆ getMsg()

```
const std::string ATEsystem_PIRIS::ErrorCluster::getMsg( ) const
```

◆ **getOk()**

```
const bool ATEsystem_PIRIS::ErrorCluster::getOk( ) const
```

◆ ToString()

```
std::string ATEsystem_PIRIS::ErrorCluster::ToString( ) const
```

Friends And Related Function Documentation

◆ operator<<

```
std::ostream& operator<< ( std::ostream & strm,  
                           const ErrorCluster & err  
                         )
```

friend

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

- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/include/[ATEsystem.PIRIS.h](#)
- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/src/[ATEsystem.PIRIS.cpp](#)

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS

Factory

[Public Member Functions](#) |

[Static Public Member Functions](#) |

[List of all members](#)

ATEsystem_PIRIS::Factory Class Reference

[More...](#)

```
#include <ATEsystem.PIRIS.h>
```

Public Member Functions

Factory (Factory const &)=delete

Factory & operator= (Factory const &)=delete

~Factory ()

DevID_t CreateDevice (Mode mode=Mode::ETHERNET)

DevID_t CreateDevice (const std::string &name, Mode mode)

IDevice * GetDeviceInstance (DevID_t id)

std::string GetDeviceName (DevID_t id)

Status RemoveDevice (DevID_t id)

Static Public Member Functions

static **Factory** * **getInstance** ()

Detailed Description

Factory class for manage PIRIS unique instances. Driver supports multiple opened PIRIS devices, that can be easily manipulated by unique DevID and IPiris Smart Pointer.

Constructor & Destructor Documentation

◆ Factory()

ATEsystem_PIRIS::Factory::Factory (**Factory** const &)

delete

◆ ~Factory()

ATEsystem_PIRIS::Factory::~Factory ()

inline

Member Function Documentation

◆ CreateDevice() [1/2]

DevID_t

ATEsystem_PIRIS::Factory::CreateDevice (**Mode mode** = Mode::ETHER

Create a PIRIS device with random string name. Usually first method to call.

Parameters

mode Mode - PIRIS device product type (SERIAL/ETHERNET)

Returns

Device unique id

◆ CreateDevice() [2/2]

DevID_t

```
ATEsystem_PIRIS::Factory::CreateDevice ( const std::string & name,  
                                         Mode                mode  
                                         )
```

Create a PIRIS device with specific string name. Usually first method to call.

Parameters

name string name of created object

Parameters

mode PIRIS device product type (SERIAL/ETHERNET)

Returns

PIRIS Device unique id

◆ GetDeviceInstance()

IDevice *

ATEsystem_PIRIS::Factory::GetDeviceInstance (**DevID_t id**)

Return Smart Pointer IPiris (IDevice). Usually call this when need to do any action with PIRIS.

Parameters

id PIRIS Device unique id

Returns

Smart Pointer IPiris (IDevice)

◆ GetDeviceName()

```
std::string  
ATEsystem_PIRIS::Factory::GetDeviceName ( DevID_t id )
```

Return PIRIS device name, either random or specified string.

Parameters

id PIRIS Device unique id

Returns

PIRIS device string name

◆ getInstance()

static **Factory***

ATEsystem_PIRIS::Factory::getInstance

()

inline

static

Return instance to singleton **Factory** class.

Returns

Factory class pointer

◆ operator=()

Factory&
ATEsystem_PIRIS::Factory::operator= (**Factory** const &) delete

◆ RemoveDevice()

Status ATESystem_PIRIS::Factory::RemoveDevice (**DevID_t id**)

Remove PIRIS device. Usually call this, when driver is used to manipulate multiple PIRIS devices.

Parameters

id PIRIS Device unique id

Returns

Status class

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

- ATESystem.PIRIS-driver/ATESystem.PIRIS-driver-Shared/include/[ATESystem.PIRIS.h](#)
- ATESystem.PIRIS-driver/ATESystem.PIRIS-driver-Shared/src/[ATESystem.PIRIS.cpp](#)

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS

FocusZoomIris

[Public Member Functions](#) | [Friends](#) |
[List of all members](#)

ATEsystem_PIRIS::FocusZoomIris< T > Class Template Reference

[More...](#)

```
#include <ATEsystem.PIRIS.h>
```

Public Member Functions

FocusZoomIris (T focus=NULL, T
zoom=NULL, T iris=NULL, int
str_pad_len=25)

const T **getFocus** () const

const T **getZoom** () const

const T **getIris** () const

const std::tuple< T, T, T > **getFocusZoomIris** () const

std::string **ToString** () const

Friends

std::ostream & **operator<<** (std::ostream &strm, const
FocusZoomIris &vals)

Detailed Description

```
template<class T = uint16_t>
class ATEsystem_PIRIS::FocusZoomIris< T >
```

Simple Focus, Zoom, Iris container template class.

Constructor & Destructor Documentation

◆ FocusZoomIris()

```
template<class T = uint16_t>
ATEsystem_PIRIS::FocusZoomIris< T
>::FocusZoomIris( T focus = NULL,
                    T zoom = NULL,
                    T iris = NULL,
                    int str_pad_len = 25
                )
```

inline

Member Function Documentation

◆ getFocus()

```
template<class T = uint16_t>
const T ATEsystem_PIRIS::FocusZoomIris< T
>::getFocus ( ) const inline
```

◆ **getFocusZoomIris()**

```
template<class T = uint16_t>
const std::tuple<T, T, T>
ATEsystem_PIRIS::FocusZoomIris< T
>::getFocusZoomIris ( ) const inline
```

◆ getIris()

```
template<class T = uint16_t>
const T ATEsystem_PIRIS::FocusZoomIris< T
>::getIris( ) const inline
```

◆ getZoom()

```
template<class T = uint16_t>
const T ATEsystem_PIRIS::FocusZoomIris< T
>::getZoom ( ) const inline
```

◆ ToString()

```
template<class T = uint16_t>
std::string ATEsystem_PIRIS::FocusZoomIris< T >::ToString( ) const inline
```

Friends And Related Function Documentation

◆ operator<<

```
template<class T = uint16_t>
std::ostream&
operator<< ( std::ostream & strm,
            const FocusZoomIris< T > & vals
            )
friend
```

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

- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/include/**ATEsystem.PIRIS.h**

Generated by [doxygen](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS

PirisDevice

[Public Member Functions](#) | [Friends](#) |
[List of all members](#)

ATEsystem_PIRIS::PirisDevice Class Reference

[More...](#)

```
#include <ATEsystem.PIRIS.h>
```

Public Member Functions

PirisDevice (Pylon::CDeviceInfo &device)

PirisDevice (Pylon::IPylonDevice
*pylon_dev)

PirisDevice (Pylon::CInstantCamera
*camera)

PirisDevice (serial::PortInfo &serial)

Pylon::CDeviceInfo & **get_device_eth** () const

Pylon::IPylonDevice * **get_pylon_dev** () const

Pylon::CInstantCamera * **get_camera** () const

serial::PortInfo & **get_serial** () const

const **Mode** **get_mode** () const

const **PirisDeviceType** **get_type** () const

std::string **ToString** () const

Friends

std::ostream & **operator<<** (std::ostream &strm, const **PirisDevice** &dev)

Detailed Description

This class holds ref or information about unique and specific PIRIS device. Its main purpose is to server as an input to open function to overcome multiple inputs problem.

Constructor & Destructor Documentation

◆ PirisDevice() [1/4]

```
ATEsystem_PIRIS::PirisDevice::PirisDevice ( Pylon::CDeviceInfo & dev )
```

◆ PirisDevice() [2/4]

```
ATEsystem_PIRIS::PirisDevice::PirisDevice ( Pylon::IPylonDevice * pyl
```

◆ PirisDevice() [3/4]

```
ATEsystem_PIRIS::PirisDevice::PirisDevice ( Pylon::CInstantCamera *
```

◆ PirisDevice() [4/4]

```
ATEsystem_PIRIS::PirisDevice::PirisDevice ( serial::PortInfo & serial )
```

Member Function Documentation

◆ get_camera()

```
Pylon::CInstantCamera *  
ATEsystem_PIRIS::PirisDevice::get_camera ( ) const
```

◆ get_device_eth()

```
Pylon::CDeviceInfo &
ATEsystem_PIRIS::PirisDevice::get_device_eth ( ) const
```

◆ get_mode()

```
const Mode ATEsystem_PIRIS::PirisDevice::get_mode( ) const
```

◆ get_pylon_dev()

```
Pylon::IPylonDevice *  
ATEsystem_PIRIS::PirisDevice::get_pylon_dev ( ) const
```

◆ get_serial()

```
serial::PortInfo & ATEsystem_PIRIS::PirisDevice::get_serial( ) const
```

◆ **get_type()**

```
const PirisDeviceType
ATEsystem_PIRIS::PirisDevice::get_type ( ) const
```

◆ ToString()

```
std::string ATEsystem_PIRIS::PirisDevice::ToString( ) const
```

Friends And Related Function Documentation

◆ operator<<

```
std::ostream& operator<< ( std::ostream & strm,  
                           const PirisDevice & dev  
                         )
```

friend

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

- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/include/[ATEsystem.PIRIS.h](#)
- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/src/[ATEsystem.PIRIS.cpp](#)

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATESystem_PIRIS

SmartPointer

[Public Member Functions](#) | [List of all members](#)

ATEsystem_PIRIS::SmartPointer< T > Class Template Reference

[More...](#)

```
#include <ATEsystem.PIRIS.h>
```

Public Member Functions

```
SmartPointer (void) noexcept
```

```
~SmartPointer (void)
```

```
void operator= (T *pB)
```

```
operator T * (void) const
```

```
T & operator* (void) const
```

```
T & operator() (void) const
```

```
T * operator-> (void) const
```

```
bool IsValid () const throw ()
```

```
operator bool (void) const throw ()
```

```
bool operator== (T *pT) const
```

Detailed Description

```
template<class T>
class ATEsystem_PIRIS::SmartPointer< T >
```

Generic Smart Pointer class, that can be used for mem mng of any type. Implementation by Basler AG.

Constructor & Destructor Documentation

◆ SmartPointer()

```
template<class T >
ATEsystem_PIRIS::SmartPointer< T
>::SmartPointer ( void ) inline noexcept
```

◆ ~SmartPointer()

```
template<class T >
ATEsystem_PIRIS::SmartPointer< T
>::~SmartPointer ( void ) inline
```

Member Function Documentation

◆ IsValid()

```
template<class T>
bool ATEsystem_PIRIS::SmartPointer< T >::IsValid( ) const
    throw(
        )
    inline
```

◆ operator *()

```
template<class T >
T& ATEsystem_PIRIS::SmartPointer< T
>::operator *() const inline
```

◆ operator bool()

```
template<class T>
ATEsystem_PIRIS::SmartPointer< T >::operator bool ( void ) const
throw ( )
)
inline
```

◆ operator T *()

```
template<class T >
ATEsystem_PIRIS::SmartPointer< T
>::operator T *
( void ) const inline
```

◆ operator()()

```
template<class T >
T& ATEsystem_PIRIS::SmartPointer< T
>::operator() ( void ) const inline
```

◆ operator->()

```
template<class T >
T* ATEsystem_PIRIS::SmartPointer< T
>::operator-> ( void ) const inline
```

◆ operator=()

```
template<class T >
void ATEsystem_PIRIS::SmartPointer< T
>::operator= ( T * pB ) inline
```

◆ operator==()

```
template<class T >
bool ATEsystem_PIRIS::SmartPointer< T
>::operator==( T * pT ) const inline
```

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

- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/include/**ATEsystem.PIRIS.h**

Generated by **doxygen** 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS > StatusEx >

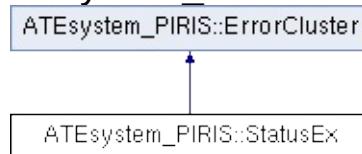
[Public Member Functions](#) | [Public Attributes](#) |
[List of all members](#)

ATEsystem_PIRIS::StatusEx Class Reference

[More...](#)

```
#include <ATEsystem.PIRIS.h>
```

Inheritance diagram for ATEsystem_PIRIS::StatusEx:



Public Member Functions

**StatusEx (Status
status=Status::GENERAL_E
ErrorCluster err=ErrorClus**

► Public Member Functions inherited from
ATEsystem_PIRIS::ErrorCluster

**ErrorCluster (std::string err_
bool override=false)**

const bool **getOk () const**

const int8_t **getErrNum () const**

const std::string **getMsg () const**

const std::tuple< bool, int8_t, std::string > **getErrorCluster () const**

std::string **ToString () const**

Public Attributes

const **Status** **status**

Detailed Description

Extended status class used to hold **ErrorCluster** and other information (version, name)

Constructor & Destructor Documentation

◆ StatusEx()

```
ATEsystem_PIRIS::StatusEx::StatusEx ( Status      status = Status::  
                                      ErrorCluster err = ErrorCluster::  
                                      )
```

Member Data Documentation

◆ status

```
const Status ATESystem_PIRIS::StatusEx::status
```

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

- ATESystem.PIRIS-driver/ATESystem.PIRIS-driver-Shared/include/[ATESystem.PIRIS.h](#)
 - ATESystem.PIRIS-driver/ATESystem.PIRIS-driver-Shared/src/[ATESystem.PIRIS.cpp](#)
-

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS

Utils

[Static Public Member Functions](#) |
[List of all members](#)

ATEsystem_PIRIS::Utils Class Reference

More...

```
#include <ATEsystem.PIRIS.h>
```

Static Public Member Functions

```
static void CpyCStr2CStr (char *dest, const char *source,  
size_t max_len)
```

```
static void CopySTLStr2CStr (char *dest, std::string source)
```

```
static std::string CopyCStr2STLStr (const char *src)
```

```
static std::string PadStr (const int total, const int minus=0, const char  
ch=' ')
```

Detailed Description

Utility class mainly for C string and C++ string handle.

Member Function Documentation

◆ CopyCStr2STLStr()

```
std::string  
ATEsystem_PIRIS::Utils::CopyCStr2STLStr ( const char * src ) static
```

◆ CopySTLStr2CStr()

```
void  
ATEsystem_PIRIS::Utils::CopySTLStr2CStr ( char * dest,  
                                         std::string source  
                                         )  
                                         static
```

◆ CpyCStr2CStr()

```
void  
ATEsystem_PIRIS::Utils::CpyCStr2CStr ( char * dest,  
                                         const char * source,  
                                         size_t max_len  
                                         )  
                                         static
```

◆ PadStr()

```
std::string  
ATEsystem_PIRIS::Utils::PadStr ( const int total,  
                                 const int minus = 0,  
                                 const char ch = ' '  
 ) static
```

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

- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/include/[ATEsystem.PIRIS.h](#)
 - ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/src/[ATEsystem.PIRIS.cpp](#)
-

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS

Version

[Public Member Functions](#) | [Friends](#) |
[List of all members](#)

ATEsystem_PIRIS::Version Class Reference

More...

```
#include <ATEsystem.PIRIS.h>
```

Public Member Functions

Version (uint16_t
major=0, uint16_t
minor=0, uint16_t
revision=0)

const uint16_t **getMajor** () const

const uint16_t **getMinor** () const

const uint16_t **getRevision** () const

const std::tuple< uint16_t, uint16_t, uint16_t > **getVersion** () const

std::string **ToString** () const

Friends

```
std::ostream & operator<< (std::ostream &strm, const Version &ver)
```

```
    bool operator== (const Version &ver1, const Version  
        &ver2)
```

```
    bool operator> (const Version &ver1, const Version  
        &ver2)
```

```
    bool operator>= (const Version &ver1, const Version  
        &ver2)
```

```
    bool operator< (const Version &ver1, const Version  
        &ver2)
```

```
    bool operator<= (const Version &ver1, const Version  
        &ver2)
```

Detailed Description

Simple version (Major, Minor, Rev) container class.

Constructor & Destructor Documentation

◆ Version()

```
ATEsystem_PIRIS::Version::Version ( uint16_t major = 0,  
                                    uint16_t minor = 0,  
                                    uint16_t revision = 0  
                                )
```

Member Function Documentation

◆ getMajor()

```
const uint16_t ATESystem_PIRIS::Version::getMajor( ) const
```

◆ getMinor()

```
const uint16_t ATESystem_PIRIS::Version::getMinor( ) const
```

◆ getRevision()

```
const uint16_t ATESystem_PIRIS::Version::getRevision( ) const
```

◆ getVersion()

```
const std::tuple< uint16_t, uint16_t, uint16_t >
ATEsystem_PIRIS::Version::getVersion ( ) const
```

◆ ToString()

```
std::string ATEsystem_PIRIS::Version::ToString( ) const
```

Friends And Related Function Documentation

◆ operator<

```
bool operator< ( const Version & ver1,  
                  const Version & ver2  
                )
```

friend

◆ operator<<

```
std::ostream& operator<< ( std::ostream & strm,  
                           const Version & ver  
                         )
```

friend

◆ operator<=

```
bool operator<= ( const Version & ver1,  
                  const Version & ver2  
                )
```

friend

◆ operator==

```
bool operator==( const Version & ver1,  
                   const Version & ver2  
)
```

friend

◆ operator>

```
bool operator> ( const Version & ver1,  
                  const Version & ver2  
              )
```

friend

◆ operator>=

```
bool operator>= ( const Version & ver1,  
                  const Version & ver2  
                )
```

friend

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

- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/include/[ATEsystem.PIRIS.h](#)
- ATEsystem.PIRIS-driver/ATEsystem.PIRIS-driver-Shared/src/[ATEsystem.PIRIS.cpp](#)

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

Class Index

[c](#) | [d](#) | [e](#) | [f](#) | [p](#) | [s](#) | [u](#) | [v](#)

c

[DataParams \(ATEsystem_PIF\)](#)

[CPirisEthernet \(ATEsystem_PIRIS\)](#)

[DataState \(ATEsystem_PIRIS\)](#)

[CPirisMain \(ATEsystem_PIRIS\)](#)

e

[CPirisSerial \(ATEsystem_PIRIS\)](#)

[ErrorCluster \(ATEsystem_PII\)](#)

d

[DataID \(ATEsystem_PIRIS\)](#)

[Factory \(ATEsystem_PIRIS\)](#)

[c](#) | [d](#) | [e](#) | [f](#) | [p](#) | [s](#) | [u](#) | [v](#)

Generated by [doxygen](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

Class Hierarchy

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

[detail level 1 2 3]

- ATEsystem_PIRIS::DataID
- ATEsystem_PIRIS::DataParams
- ATEsystem_PIRIS::DataPosition
- ATEsystem_PIRIS::DataState
- ▼ ● ATEsystem_PIRIS::ErrorCluster
 - ATEsystem_PIRIS::StatusEx
 - ATEsystem_PIRIS::Factory
 - ATEsystem_PIRIS::FocusZoomIris< T >
 - ATEsystem_PIRIS::FocusZoomIris< bool >
 - ATEsystem_PIRIS::FocusZoomIris< uint16_t >
- ▼ ● IComm
 - ▼ ● ATEsystem_PIRIS::CPirisMain
 - ATEsystem_PIRIS::CPirisEthernet
 - ATEsystem_PIRIS::CPirisSerial
- ▼ ● IDevice
 - ATEsystem_PIRIS::CPirisMain
 - ATEsystem_PIRIS::PirisDevice
 - ATEsystem_PIRIS::SmartPointer< T >
 - ATEsystem_PIRIS::Utils
 - ATEsystem_PIRIS::Version

Generated by [doxygen](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

Here is a list of all class members with links to the classes they belong to:

- C -

- close() : [ATEsystem_PIRIS::CPirisEthernet](#)
- Close() : [ATEsystem_PIRIS::CPirisMain](#)
- close() : [ATEsystem_PIRIS::CPirisSerial](#)
- CopyCStr2STLStr() : [ATEsystem_PIRIS::Utils](#)
- CopySTLStr2CStr() : [ATEsystem_PIRIS::Utils](#)
- CPirisEthernet() : [ATEsystem_PIRIS::CPirisEthernet](#)
- CPirisMain() : [ATEsystem_PIRIS::CPirisMain](#)
- CPirisSerial() : [ATEsystem_PIRIS::CPirisSerial](#)
- CpyCStr2CStr() : [ATEsystem_PIRIS::Utils](#)
- create() : [ATEsystem_PIRIS::CPirisEthernet](#) ,
[ATEsystem_PIRIS::CPirisSerial](#)
- CreateDevice() : [ATEsystem_PIRIS::Factory](#)

- d -

- DataID() : [ATEsystem_PIRIS::DataID](#)
- DataParams() : [ATEsystem_PIRIS::DataParams](#)
- DataPosition() : [ATEsystem_PIRIS::DataPosition](#)
- DataState() : [ATEsystem_PIRIS::DataState](#)
- DevHoming() : [ATEsystem_PIRIS::CPirisMain](#)
- DevReset() : [ATEsystem_PIRIS::CPirisMain](#)

- e -

- ErrorCluster() : [ATEsystem_PIRIS::ErrorCluster](#)

- f -

- Factory() : **ATEsystem_PIRIS::Factory**
- flush() : **ATEsystem_PIRIS::CPirisEthernet** ,
ATEsystem_PIRIS::CPirisSerial
- FocusZoomIris() : **ATEsystem_PIRIS::FocusZoomIris< T >**

- g -

- get_camera() : **ATEsystem_PIRIS::PirisDevice**
- get_device_eth() : **ATEsystem_PIRIS::PirisDevice**
- get_mode() : **ATEsystem_PIRIS::PirisDevice**
- get_pylon_dev() : **ATEsystem_PIRIS::PirisDevice**
- get_serial() : **ATEsystem_PIRIS::PirisDevice**
- get_type() : **ATEsystem_PIRIS::PirisDevice**
- GetDeviceInstance() : **ATEsystem_PIRIS::Factory**
- GetDeviceName() : **ATEsystem_PIRIS::Factory**
- getErrNum() : **ATEsystem_PIRIS::ErrorCluster**
- getErrorCluster() : **ATEsystem_PIRIS::ErrorCluster**
- getFocus() : **ATEsystem_PIRIS::FocusZoomIris< T >**
- getFocusZoomIris() : **ATEsystem_PIRIS::FocusZoomIris< T >**
- GetFwPollSupport() : **ATEsystem_PIRIS::CPirisMain**
- getInstance() : **ATEsystem_PIRIS::Factory**
- getIris() : **ATEsystem_PIRIS::FocusZoomIris< T >**
- getMajor() : **ATEsystem_PIRIS::Version**
- getMinor() : **ATEsystem_PIRIS::Version**
- getMsg() : **ATEsystem_PIRIS::ErrorCluster**
- getOk() : **ATEsystem_PIRIS::ErrorCluster**
- getRevision() : **ATEsystem_PIRIS::Version**
- GetVerboseLevel() : **ATEsystem_PIRIS::CPirisMain**
- getVersion() : **ATEsystem_PIRIS::Version**
- getZoom() : **ATEsystem_PIRIS::FocusZoomIris< T >**

- i -

- ir_filter : **ATEsystem_PIRIS::DataPosition**
- ir_present : **ATEsystem_PIRIS::DataParams**
- IsValid() : **ATEsystem_PIRIS::SmartPointer< T >**

- l -

- lens : **ATEsystem_PIRIS::DataParams**

- m -

- max_value : [ATEsystem_PIRIS::DataParams](#)
- motors_busy : [ATEsystem_PIRIS::DataState](#)

- n -

- name : [ATEsystem_PIRIS::DataID](#)

- o -

- open() : [ATEsystem_PIRIS::CPirisEthernet](#)
- Open() : [ATEsystem_PIRIS::CPirisMain](#)
- open() : [ATEsystem_PIRIS::CPirisSerial](#)
- operator *() : [ATEsystem_PIRIS::SmartPointer< T >](#)
- operator bool() : [ATEsystem_PIRIS::SmartPointer< T >](#)
- operator T *() : [ATEsystem_PIRIS::SmartPointer< T >](#)
- operator()() : [ATEsystem_PIRIS::SmartPointer< T >](#)
- operator->() : [ATEsystem_PIRIS::SmartPointer< T >](#)
- operator< : [ATEsystem_PIRIS::Version](#)
- operator<< : [ATEsystem_PIRIS::ErrorCluster](#) ,
[ATEsystem_PIRIS::FocusZoomIris< T >](#) ,
[ATEsystem_PIRIS::PirisDevice](#) , [ATEsystem_PIRIS::Version](#)
- operator<= : [ATEsystem_PIRIS::Version](#)
- operator=() : [ATEsystem_PIRIS::Factory](#) ,
[ATEsystem_PIRIS::SmartPointer< T >](#)
- operator==() : [ATEsystem_PIRIS::SmartPointer< T >](#) ,
[ATEsystem_PIRIS::Version](#)
- operator> : [ATEsystem_PIRIS::Version](#)
- operator>= : [ATEsystem_PIRIS::Version](#)

- p -

- PadStr() : [ATEsystem_PIRIS::Utils](#)
- PirisDevice() : [ATEsystem_PIRIS::PirisDevice](#)
- position : [ATEsystem_PIRIS::DataPosition](#)

- r -

- `read()` : `ATEsystem_PIRIS::CPirisEthernet` ,
`ATEsystem_PIRIS::CPirisSerial`
- `ReadID()` : `ATEsystem_PIRIS::CPirisMain`
- `ReadParams()` : `ATEsystem_PIRIS::CPirisMain`
- `ReadPosition()` : `ATEsystem_PIRIS::CPirisMain`
- `ReadState()` : `ATEsystem_PIRIS::CPirisMain`
- `remove()` : `ATEsystem_PIRIS::CPirisEthernet` ,
`ATEsystem_PIRIS::CPirisSerial`
- `RemoveDevice()` : `ATEsystem_PIRIS::Factory`
- `rx_timeout` : `ATEsystem_PIRIS::CPirisMain`

- S -

- `scan()` : `ATEsystem_PIRIS::CPirisEthernet` ,
`ATEsystem_PIRIS::CPirisSerial`
- `ScanEthernet()` : `ATEsystem_PIRIS::CPirisMain`
- `ScanSerial()` : `ATEsystem_PIRIS::CPirisMain`
- `sens_present` : `ATEsystem_PIRIS::DataParams`
- `SetAbsolute()` : `ATEsystem_PIRIS::CPirisMain`
- `SetRelative()` : `ATEsystem_PIRIS::CPirisMain`
- `SetVerboseLevel()` : `ATEsystem_PIRIS::CPirisMain`
- `SmartPointer()` : `ATEsystem_PIRIS::SmartPointer< T >`
- `state` : `ATEsystem_PIRIS::DataState`
- `state_get()` : `ATEsystem_PIRIS::CPirisMain`
- `state_is()` : `ATEsystem_PIRIS::CPirisMain`
- `state_reset()` : `ATEsystem_PIRIS::CPirisMain`
- `state_set()` : `ATEsystem_PIRIS::CPirisMain`
- `status` : `ATEsystem_PIRIS::StatusEx`
- `StatusEx()` : `ATEsystem_PIRIS::StatusEx`

- t -

- `ToString()` : `ATEsystem_PIRIS::ErrorCluster` ,
`ATEsystem_PIRIS::FocusZoomIris< T >` ,
`ATEsystem_PIRIS::PirisDevice` , `ATEsystem_PIRIS::Version`

- v -

- `verbose_is()` : `ATEsystem_PIRIS::CPirisMain`
- `verbose_set()` : `ATEsystem_PIRIS::CPirisMain`

- version : [ATEsystem_PIRIS::DataID](#)
- Version() : [ATEsystem_PIRIS::Version](#)

- W -

- write() : [ATEsystem_PIRIS::CPirisEthernet](#) ,
[ATEsystem_PIRIS::CPirisSerial](#)

- ~ -

- ~CPirisEthernet() : [ATEsystem_PIRIS::CPirisEthernet](#)
- ~CPirisMain() : [ATEsystem_PIRIS::CPirisMain](#)
- ~CPirisSerial() : [ATEsystem_PIRIS::CPirisSerial](#)
- ~Factory() : [ATEsystem_PIRIS::Factory](#)
- ~SmartPointer() : [ATEsystem_PIRIS::SmartPointer< T >](#)

ATEsystem.PIRIS-driver



1.0.1
ATEsystem.PIRIS-driver

- C -

- close() : **ATEsystem_PIRIS::CPirisEthernet**
- Close() : **ATEsystem_PIRIS::CPirisMain**
- close() : **ATEsystem_PIRIS::CPirisSerial**
- CopyCStr2STLStr() : **ATEsystem_PIRIS::Utils**
- CopySTLStr2CStr() : **ATEsystem_PIRIS::Utils**
- CPirisEthernet() : **ATEsystem_PIRIS::CPirisEthernet**
- CPirisMain() : **ATEsystem_PIRIS::CPirisMain**
- CPirisSerial() : **ATEsystem_PIRIS::CPirisSerial**
- CpyCStr2CStr() : **ATEsystem_PIRIS::Utils**
- create() : **ATEsystem_PIRIS::CPirisEthernet** ,
ATEsystem_PIRIS::CPirisSerial
- CreateDevice() : **ATEsystem_PIRIS::Factory**

- d -

- DataID() : **ATEsystem_PIRIS::DataID**
- DataParams() : **ATEsystem_PIRIS::DataParams**
- DataPosition() : **ATEsystem_PIRIS::DataPosition**
- DataState() : **ATEsystem_PIRIS::DataState**
- DevHoming() : **ATEsystem_PIRIS::CPirisMain**
- DevReset() : **ATEsystem_PIRIS::CPirisMain**

- e -

- ErrorCluster() : **ATEsystem_PIRIS::ErrorCluster**

- f -

- Factory() : **ATEsystem_PIRIS::Factory**

- flush() : **ATEsystem_PIRIS::CPirisEthernet** ,
ATEsystem_PIRIS::CPirisSerial
- FocusZoomIris() : **ATEsystem_PIRIS::FocusZoomIris< T >**

- g -

- get_camera() : **ATEsystem_PIRIS::PirisDevice**
- get_device_eth() : **ATEsystem_PIRIS::PirisDevice**
- get_mode() : **ATEsystem_PIRIS::PirisDevice**
- get_pylon_dev() : **ATEsystem_PIRIS::PirisDevice**
- get_serial() : **ATEsystem_PIRIS::PirisDevice**
- get_type() : **ATEsystem_PIRIS::PirisDevice**
- GetDeviceInstance() : **ATEsystem_PIRIS::Factory**
- GetDeviceName() : **ATEsystem_PIRIS::Factory**
- getErrNum() : **ATEsystem_PIRIS::ErrorCluster**
- getErrorCluster() : **ATEsystem_PIRIS::ErrorCluster**
- getFocus() : **ATEsystem_PIRIS::FocusZoomIris< T >**
- getFocusZoomIris() : **ATEsystem_PIRIS::FocusZoomIris< T >**
- GetFwPollSupport() : **ATEsystem_PIRIS::CPirisMain**
- getInstance() : **ATEsystem_PIRIS::Factory**
- getIris() : **ATEsystem_PIRIS::FocusZoomIris< T >**
- getMajor() : **ATEsystem_PIRIS::Version**
- getMinor() : **ATEsystem_PIRIS::Version**
- getMsg() : **ATEsystem_PIRIS::ErrorCluster**
- getOk() : **ATEsystem_PIRIS::ErrorCluster**
- getRevision() : **ATEsystem_PIRIS::Version**
- GetVerboseLevel() : **ATEsystem_PIRIS::CPirisMain**
- getVersion() : **ATEsystem_PIRIS::Version**
- getZoom() : **ATEsystem_PIRIS::FocusZoomIris< T >**

- i -

- IsValid() : **ATEsystem_PIRIS::SmartPointer< T >**

- O -

- open() : **ATEsystem_PIRIS::CPirisEthernet**
- Open() : **ATEsystem_PIRIS::CPirisMain**
- open() : **ATEsystem_PIRIS::CPirisSerial**
- operator *() : **ATEsystem_PIRIS::SmartPointer< T >**

- operator bool() : **ATEsystem_PIRIS::SmartPointer< T >**
- operator T *() : **ATEsystem_PIRIS::SmartPointer< T >**
- operator()() : **ATEsystem_PIRIS::SmartPointer< T >**
- operator->() : **ATEsystem_PIRIS::SmartPointer< T >**
- operator=() : **ATEsystem_PIRIS::Factory** ,
ATEsystem_PIRIS::SmartPointer< T >
- operator==() : **ATEsystem_PIRIS::SmartPointer< T >**

- p -

- PadStr() : **ATEsystem_PIRIS::Utils**
- PirisDevice() : **ATEsystem_PIRIS::PirisDevice**

- r -

- read() : **ATEsystem_PIRIS::CPirisEthernet** ,
ATEsystem_PIRIS::CPirisSerial
- ReadID() : **ATEsystem_PIRIS::CPirisMain**
- ReadParams() : **ATEsystem_PIRIS::CPirisMain**
- ReadPosition() : **ATEsystem_PIRIS::CPirisMain**
- ReadState() : **ATEsystem_PIRIS::CPirisMain**
- remove() : **ATEsystem_PIRIS::CPirisEthernet** ,
ATEsystem_PIRIS::CPirisSerial
- RemoveDevice() : **ATEsystem_PIRIS::Factory**

- s -

- scan() : **ATEsystem_PIRIS::CPirisEthernet** ,
ATEsystem_PIRIS::CPirisSerial
- ScanEthernet() : **ATEsystem_PIRIS::CPirisMain**
- ScanSerial() : **ATEsystem_PIRIS::CPirisMain**
- SetAbsolute() : **ATEsystem_PIRIS::CPirisMain**
- SetRelative() : **ATEsystem_PIRIS::CPirisMain**
- SetVerboseLevel() : **ATEsystem_PIRIS::CPirisMain**
- SmartPointer() : **ATEsystem_PIRIS::SmartPointer< T >**
- state_get() : **ATEsystem_PIRIS::CPirisMain**
- state_is() : **ATEsystem_PIRIS::CPirisMain**
- state_reset() : **ATEsystem_PIRIS::CPirisMain**
- state_set() : **ATEsystem_PIRIS::CPirisMain**
- StatusEx() : **ATEsystem_PIRIS::StatusEx**

- t -

- ToString() : [ATEsystem_PIRIS::ErrorCluster](#) ,
[ATEsystem_PIRIS::FocusZoomIris< T >](#) ,
[ATEsystem_PIRIS::PirisDevice](#) , [ATEsystem_PIRIS::Version](#)

- v -

- verbose_is() : [ATEsystem_PIRIS::CPirisMain](#)
- verbose_set() : [ATEsystem_PIRIS::CPirisMain](#)
- Version() : [ATEsystem_PIRIS::Version](#)

- w -

- write() : [ATEsystem_PIRIS::CPirisEthernet](#) ,
[ATEsystem_PIRIS::CPirisSerial](#)

- ~ -

- ~CPirisEthernet() : [ATEsystem_PIRIS::CPirisEthernet](#)
- ~CPirisMain() : [ATEsystem_PIRIS::CPirisMain](#)
- ~CPirisSerial() : [ATEsystem_PIRIS::CPirisSerial](#)
- ~Factory() : [ATEsystem_PIRIS::Factory](#)
- ~SmartPointer() : [ATEsystem_PIRIS::SmartPointer< T >](#)

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

- ir_filter : `ATEsystem_PIRIS::DataPosition`
 - ir_present : `ATEsystem_PIRIS::DataParams`
 - lens : `ATEsystem_PIRIS::DataParams`
 - max_value : `ATEsystem_PIRIS::DataParams`
 - motors_busy : `ATEsystem_PIRIS::DataState`
 - name : `ATEsystem_PIRIS::DataID`
 - position : `ATEsystem_PIRIS::DataPosition`
 - rx_timeout : `ATEsystem_PIRIS::CPirisMain`
 - sens_present : `ATEsystem_PIRIS::DataParams`
 - state : `ATEsystem_PIRIS::DataState`
 - status : `ATEsystem_PIRIS::StatusEx`
 - version : `ATEsystem_PIRIS::DataID`
-

Generated by doxygen 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

- operator< : `ATEsystem_PIRIS::Version`
 - operator<< : `ATEsystem_PIRIS::ErrorCluster` ,
`ATEsystem_PIRIS::FocusZoomIris< T >` ,
`ATEsystem_PIRIS::PirisDevice` , `ATEsystem_PIRIS::Version`
 - operator<= : `ATEsystem_PIRIS::Version`
 - operator== : `ATEsystem_PIRIS::Version`
 - operator> : `ATEsystem_PIRIS::Version`
 - operator>= : `ATEsystem_PIRIS::Version`
-

Generated by doxygen 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

File List

Here is a list of all files with brief descriptions:

[detail level [1](#) [2](#) [3](#) [4](#)]

▼ ATEsystem.PIRIS-driver	
▼ ATEsystem.PIRIS-driver-Shared	
▼ include	
ATEsystem.PIRIS.h	
ATEsystem.PIRIS_ethernet.h	
ATEsystem.PIRIS_serial.h	
▼ src	
ATEsystem.PIRIS.cpp	
ATEsystem.PIRIS_ethernet.cpp	
ATEsystem.PIRIS_serial.cpp	
example.cpp	
example_ethernet.cpp	
example_serial.cpp	
doxygen_doc	

Generated by [doxygen](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver Directory Reference

Directories

directory **ATEsystem.PIRIS-driver-Shared**

directory **doxygen_doc**

Generated by **doxygen** 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver-Shared

ATEsystem.PIRIS-driver-Shared Directory Reference

Directories

directory **include**

directory **src**

Files

file [**example.cpp**](#)

file [**example_ethernet.cpp**](#)

file [**example_serial.cpp**](#)

Generated by [**doxygen**](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver > ATEsystem.PIRIS-driver-Shared > include >

include Directory Reference

Files

file [**ATEsystem.PIRIS.h**](#) [code]

file [**ATEsystem.PIRIS_ethernet.h**](#) [code]

file [**ATEsystem.PIRIS_serial.h**](#) [code]

Generated by [**doxygen**](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver-Shared

include

[Classes](#) | [Namespaces](#) | [Macros](#) | [Typedefs](#) |
[Enumerations](#) | [Functions](#) | [Variables](#)

ATEsystem.PIRIS.h File Reference

```
#include <string> #include <map>
#include <vector>
#include <list>
#include "pylon/PylonIncludes.h"
#include "serial.h"
```

[Go to the source code of this file.](#)

Classes

class **ATEsystem_PIRIS::Utils**

class **ATEsystem_PIRIS::PirisDevice**

class **ATEsystem_PIRIS::ErrorCluster**

class **ATEsystem_PIRIS::Version**

class **ATEsystem_PIRIS::FocusZoomIris< T >**

class **ATEsystem_PIRIS::StatusEx**

class **ATEsystem_PIRIS::DataID**

class **ATEsystem_PIRIS::DataPosition**

class **ATEsystem_PIRIS::DataParams**

class **ATEsystem_PIRIS::DataState**

class **ATEsystem_PIRIS::CPirisMain**

class **ATEsystem_PIRIS::SmartPointer< T >**

class **ATEsystem_PIRIS::Factory**

Namespaces

ATEsystem_PIRIS

Macros

```
#define OK "OK"
```

```
#define ERR0 "Command was not recognized."
```

```
#define ERR1 "Wrong input parameter."
```

```
#define ERR2 "Device is not initialized."
```

```
#define ERR3 "Internal driver circuit is in error state (overheat,  
undervoltage)."
```

```
#define ERR4 "Wrong lens type."
```

```
#define ERR5 "Damaged or missing rear sensor."
```

```
#define ERR6 "Step generator timer overflow."
```

```
#define ERR7 "Device is busy."
```

```
#define ERR_UN "Unknown error."
```

```
#define ERR_PA "Parse answer failed."
```

```
#define TXT_SUCCESS "Success"
```

```
#define TXT_GENERAL_ERROR "General Error"
```

```
#define TXT_OPEN_FAILED "Open failed"
```

```
#define TXT_CLOSE_FAILED "Close failed"
```

```
#define TXT_ALREADY_OPEN "Already open"
```

```
#define TXT_DEVICE_IS_BUSY "Device is busy"
```

```
#define TXT_DEVICE_IS_CLOSED "Device is closed"

#define TXT_INVALID_DEVICE "Invalid device"

#define TXT_WRITE_ERROR "Write error"

#define TXT_READ_ERROR "Read error"

#define TXT_SCAN_FAILED "Scan failed"

#define TXT_INTERNAL_ERROR "Internal error"

#define TXT_RX_TIMEOUT "Receive timeout"

#define TXT_DEV_SET_FAILED "Device set failed"

#define TXT_WRONG_MODE "Wrong mode"

#define TXT_DEVICE_NOT_EXIST "Device does not exist"

#define TXT_PARSE_ANSWER_FAIL "Parse answer failed"

#define TXT_ETHERNET_DEVICE "Ethernet device"

#define TXT_ETHERNET_PYLON "Ethernet pylon (camera  
overiden)"

#define TXT_ETHERNET_CAMERA "Ethernet camera (camera  
overiden)"

#define TXT_SERIAL "Serial"

#define TXT_SUCCESS2 "success"

#define TXT_FAILED "failed"
```

```
#define TXT_NAME "name"

#define TXT_VERSION "version"

#define TXT_FOCUS "focus"

#define TXT_ZOOM "zoom"

#define TXT_IRIS "iris"

#define TXT_LENS "lens"

#define TXT_IR_FILTER "ir_filter"

#define TXT_IR_PRESENT "ir_present"

#define TXT_SENS_PRESENT "sens_present"

#define TXT_BUSY "busy"

#define TXT_TYPE "type"

#define CMD_READ_ID "IDN"

#define CMD_READ_POS "GP"

#define CMD_READ_TYPE "GT"

#define CMD_READ_STATE "GS"

#define CMD_RESET "RST"

#define CMD_HOMING "INI"

#define CMD_SET_ABS "SETA:"

#define CMD_SET_REL "SETR:"
```

```
#define CMD_SUFFIX "\r\n"

#define RESP_FOCUS "F%hu"

#define RESP_ZOOM "Z%hu"

#define RESP_IRIS "P%hu"

#define TRUE_VAL '1'

#define DELIM ":""

#define LEN_CMD_READ_ID 3

#define LEN_CMD_READ_POS 5

#define LEN_CMD_READ_TYPE 7

#define LEN_CMD_READ_STATE1 2

#define LEN_CMD_READ_STATE2 3

#define LEN_CMD_RESET 0

#define LEN_CMD_HOMING 1

#define LEN_CMD_SET_ABS 1

#define LEN_CMD_SET_REL 1

#define FW_MIN_STAT_POLL 1, 7, 2

#define RX_TIMEOUT_OLD_FW 20000

#define RX_TIMEOUT_NEW_FW 1000
```

```
#define MSG(x) x CMD_SUFFIX
```

```
#define STR_MAX_SIZE 200
```

```
#define GET_VAR_NAME(Variable) (#Variable)
```

Typedefs

```
typedef int16_t ATEsystem_PIRIS::Status_t
```

```
typedef int16_t ATEsystem_PIRIS::DevID_t
```

```
typedef SmartPointer< IDDevice > ATEsystem_PIRIS::IPiris
```

```
typedef IDDevice *(* ATEsystem_PIRIS::CreateDeviceFn)(void)
```

Enumerations

```
enum ATEsystem_PIRIS::Mode : uint8_t {  
    ATEsystem_PIRIS::SERIAL = 0,  
    ATEsystem_PIRIS::ETHERNET = 1 }
```

```
enum ATEsystem_PIRIS::Status : int16_t {  
    ATEsystem_PIRIS::SUCCESS = 0,  
    ATEsystem_PIRIS::GENERAL_ERROR = -1,  
    ATEsystem_PIRIS::OPEN_FAILED = -2,  
    ATEsystem_PIRIS::CLOSE_FAILED = -3,  
    ATEsystem_PIRIS::ALREADY_OPEN = -4,  
    ATEsystem_PIRIS::DEVICE_IS_BUSY = -5,  
    ATEsystem_PIRIS::DEVICE_IS_CLOSED = -6,  
    ATEsystem_PIRIS::INVALID_DEVICE = -7,  
    ATEsystem_PIRIS::WRITE_ERROR = -8,  
    ATEsystem_PIRIS::READ_ERROR = -9,  
    ATEsystem_PIRIS::SCAN_FAILED = -10,  
    ATEsystem_PIRIS::INTERNAL_ERROR = -11,  
    ATEsystem_PIRIS::RX_TIMEOUT = -12,  
    ATEsystem_PIRIS::DEV_SET_FAILED = -13,  
    ATEsystem_PIRIS::WRONG_MODE = -14,  
    ATEsystem_PIRIS::DEVICE_NOT_EXIST = -15,  
    ATEsystem_PIRIS::PARSE_ANSWER_FAIL = -16,  
    ATEsystem_PIRIS::UNDEFINED_ERROR = -20  
}
```

```
enum ATEsystem_PIRIS::State : int8_t {  
    ATEsystem_PIRIS::UNKNOWN = 0,  
    ATEsystem_PIRIS::CLOSED = 1, ATEsystem_PIRIS::BUSY  
    = 2, ATEsystem_PIRIS::READY = 3 }
```

```
enum ATEsystem_PIRIS::PirisDeviceType : uint8_t {  
    ATEsystem_PIRIS::ETHERNET_DEVICE = 0,  
    ATEsystem_PIRIS::ETHERNET_PYTHON = 1,  
    ATEsystem_PIRIS::ETHERNET_CAMERA = 2,  
    ATEsystem_PIRIS::SERIAL_UART = 3 }
```

```
enum ATEsystem_PIRIS::VerboseLevel : uint8_t {  
    ATEsystem_PIRIS::NONE = 0, ATEsystem_PIRIS::BASIC =  
    1, ATEsystem_PIRIS::FULL = 2 }
```

```
enum ATEsystem_PIRIS::YesNoNA : uint8_t {  
    ATEsystem_PIRIS::NOT_AVAILABLE = 0,  
    ATEsystem_PIRIS::YES = 1, ATEsystem_PIRIS::NO = 2 }
```

Functions

std::ostream & [**ATEsystem_PIRIS::opera**](#)
(std::ostream &os, const St

std::ostream & [**ATEsystem_PIRIS::opera**](#)
(std::ostream &os, const
PirisDeviceType type)

virtual Status [**ATEsystem_PIRIS::Open**](#)
PirisDevice &dev, VerboseL
verbose=VerboseLevel::NC

virtual Status [**ATEsystem_PIRIS::Close**](#)

virtual std::tuple< StatusEx, DataID > [**ATEsystem_PIRIS::ReadI**](#)

virtual std::tuple< StatusEx, DataPosition > [**ATEsystem_PIRIS::ReadP**](#)

virtual std::tuple< StatusEx, DataParams > [**ATEsystem_PIRIS::ReadD**](#)

virtual std::tuple< StatusEx, DataState > [**ATEsystem_PIRIS::ReadS**](#)

virtual StatusEx [**ATEsystem_PIRIS::DevRe**](#)

virtual StatusEx [**ATEsystem_PIRIS::DevH**](#)

virtual StatusEx [**ATEsystem_PIRIS::SetAk**](#)
(uint16_t focus=0, uint16_t
uint16_t iris=0, bool ir_filter

virtual StatusEx [**ATEsystem_PIRIS::SetAk**](#)
(const FocusZoomIris< uint
&values, bool ir_filter=false

virtual StatusEx [**ATEsystem_PIRIS::SetRe**](#)
(int16_t focus=0, int16_t zo

int16_t iris=0)=0

virtual StatusEx **ATEsystem_PIRIS::SetRe**
FocusZoomIris< int16_t > {

virtual YesNoNA **ATEsystem_PIRIS::GetFv**
()=0

virtual VerboseLevel **ATEsystem_PIRIS::GetVe**
()=0

virtual void **ATEsystem_PIRIS::SetVe**
(VerboseLevel level)=0

virtual **ATEsystem_PIRIS::~IDev**

virtual Status **ATEsystem_PIRIS::close**
ignore_err=false)=0

virtual Status **ATEsystem_PIRIS::write**
std::string data)=0

virtual Status **ATEsystem_PIRIS::read** (

&data)=0

virtual Status **ATEsystem_PIRIS::flush**
ignore_err=false)=0

virtual void **ATEsystem_PIRIS::remov**

virtual **ATEsystem_PIRIS::~ICon**

Variables

Interface **ATEsystem_PIRIS::IDevice**

Interface **ATEsystem_PIRIS::IComm**

Macro Definition Documentation

◆ CMD_HOMING

```
#define CMD_HOMING "INI"
```

◆ CMD_READ_ID

```
#define CMD_READ_ID "IDN"
```

◆ CMD_READ_POS

```
#define CMD_READ_POS "GP"
```

◆ CMD_READ_STATE

```
#define CMD_READ_STATE "GS"
```

◆ CMD_READ_TYPE

```
#define CMD_READ_TYPE "GT"
```

◆ CMD_RESET

```
#define CMD_RESET "RST"
```

◆ CMD_SET_ABS

```
#define CMD_SET_ABS "SETA:"
```

◆ CMD_SET_REL

```
#define CMD_SET_REL "SETR:"
```

◆ CMD_SUFFIX

```
#define CMD_SUFFIX "\r\n"
```

◆ DELIM

```
#define DELIM ";"
```

◆ ERR0

```
#define ERR0 "Command was not recognized."
```

◆ ERR1

```
#define ERR1 "Wrong input parameter."
```

◆ ERR2

```
#define ERR2 "Device is not initialized."
```

◆ ERR3

```
#define ERR3 "Internal driver circuit is in error state (overheat,  
undervoltage)."
```

◆ ERR4

```
#define ERR4 "Wrong lens type."
```

◆ ERR5

```
#define ERR5 "Damaged or missing rear sensor."
```

◆ ERR6

```
#define ERR6 "Step generator timer overflow."
```

◆ ERR7

```
#define ERR7 "Device is busy."
```

◆ ERR_PA

```
#define ERR_PA "Parse answer failed."
```

◆ **ERR_UN**

```
#define ERR_UN "Unknown error."
```

◆ FW_MIN_STAT_POLL

```
#define FW_MIN_STAT_POLL 1, 7, 2
```

◆ GET_VAR_NAME

```
#define GET_VAR_NAME( Variable ) (#Variable)
```

◆ LEN_CMD_HOMING

```
#define LEN_CMD_HOMING 1
```

◆ LEN_CMD_READ_ID

```
#define LEN_CMD_READ_ID 3
```

◆ LEN_CMD_READ_POS

```
#define LEN_CMD_READ_POS 5
```

◆ LEN_CMD_READ_STATE1

```
#define LEN_CMD_READ_STATE1 2
```

◆ LEN_CMD_READ_STATE2

```
#define LEN_CMD_READ_STATE2 3
```

◆ LEN_CMD_READ_TYPE

```
#define LEN_CMD_READ_TYPE 7
```

◆ LEN_CMD_RESET

```
#define LEN_CMD_RESET 0
```

◆ LEN_CMD_SET_ABS

```
#define LEN_CMD_SET_ABS 1
```

◆ LEN_CMD_SET_REL

```
#define LEN_CMD_SET_REL 1
```

◆ MSG

```
#define MSG( x ) x CMD_SUFFIX
```

◆ OK

```
#define OK "OK"
```

◆ RESP_FOCUS

```
#define RESP_FOCUS "F%hu"
```

◆ RESP_IRIS

```
#define RESP_IRIS "P%hu"
```

◆ RESP_ZOOM

```
#define RESP_ZOOM "Z%hu"
```

◆ RX_TIMEOUT_NEW_FW

```
#define RX_TIMEOUT_NEW_FW 1000
```

◆ RX_TIMEOUT_OLD_FW

```
#define RX_TIMEOUT_OLD_FW 20000
```

◆ STR_MAX_SIZE

```
#define STR_MAX_SIZE 200
```

◆ TRUE_VAL

```
#define TRUE_VAL '1'
```

◆ TXT_ALREADY_OPEN

```
#define TXT_ALREADY_OPEN "Already open"
```

◆ TXT_BUSY

```
#define TXT_BUSY "busy"
```

◆ TXT_CLOSE_FAILED

```
#define TXT_CLOSE_FAILED "CLose failed"
```

◆ TXT_DEV_SET_FAILED

```
#define TXT_DEV_SET_FAILED "Device set failed"
```

◆ **TXT_DEVICE_IS_BUSY**

```
#define TXT_DEVICE_IS_BUSY "Device is busy"
```

◆ **TXT_DEVICE_IS_CLOSED**

```
#define TXT_DEVICE_IS_CLOSED "Device is closed"
```

◆ **TXT_DEVICE_NOT_EXIST**

```
#define TXT_DEVICE_NOT_EXIST "Device does not exist"
```

◆ TXT_ETHERNET_CAMERA

```
#define TXT_ETHERNET_CAMERA "Ethernet camera (camera  
overiden)"
```

◆ TXT_ETHERNET_DEVICE

```
#define TXT_ETHERNET_DEVICE "Ethernet device"
```

◆ TXT_ETHERNET_PYTHON

```
#define TXT_ETHERNET_PYTHON "Ethernet pylon (camera  
overiden)"
```

◆ TXT_FAILED

```
#define TXT_FAILED "failed"
```

◆ TXT_FOCUS

```
#define TXT_FOCUS "focus"
```

◆ TXT_GENERAL_ERROR

```
#define TXT_GENERAL_ERROR "General Error"
```

◆ TXT_INTERNAL_ERROR

```
#define TXT_INTERNAL_ERROR "Internal error"
```

◆ TXT_INVALID_DEVICE

```
#define TXT_INVALID_DEVICE "Invalid device"
```

◆ TXT_IR_FILTER

```
#define TXT_IR_FILTER "ir_filter"
```

◆ TXT_IR_PRESENT

```
#define TXT_IR_PRESENT "ir_present"
```

◆ TXT_IRIS

```
#define TXT_IRIS "iris"
```

◆ TXT_LENS

```
#define TXT_LENS "lens"
```

◆ TXT_NAME

```
#define TXT_NAME "name"
```

◆ TXT_OPEN_FAILED

```
#define TXT_OPEN_FAILED "Open failed"
```

◆ TXT_PARSE_ANSWER_FAIL

```
#define TXT_PARSE_ANSWER_FAIL "Parse answer failed"
```

◆ TXT_READ_ERROR

```
#define TXT_READ_ERROR "Read error"
```

◆ TXT_RX_TIMEOUT

```
#define TXT_RX_TIMEOUT "Receive timeout"
```

◆ TXT_SCAN_FAILED

```
#define TXT_SCAN_FAILED "Scan failed"
```

◆ TXT_SENS_PRESENT

```
#define TXT_SENS_PRESENT "sens_present"
```

◆ TXT_SERIAL

```
#define TXT_SERIAL "Serial"
```

◆ TXT_SUCCESS

```
#define TXT_SUCCESS "Success"
```

◆ TXT_SUCCESS2

```
#define TXT_SUCCESS2 "success"
```

◆ TXT_TYPE

```
#define TXT_TYPE "type"
```

◆ TXT_VERSION

```
#define TXT_VERSION "version"
```

◆ TXT_WRITE_ERROR

```
#define TXT_WRITE_ERROR "Write error"
```

◆ TXT_WRONG_MODE

```
#define TXT_WRONG_MODE "Wrong mode"
```

◆ TXT_ZOOM

```
#define TXT_ZOOM "zoom"
```

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver-Shared

include

[Classes](#) | [Namespaces](#) | [Macros](#)

ATEsystem.PIRIS_ethernet.h File Reference

```
#include <string> #include <vector>
#include "pylon/PylonIncludes.h"
#include "ATEsystem.PIRIS.h"
```

[Go to the source code of this file.](#)

Classes

class **ATEsystem_PIRIS::CPirisEthernet**

Namespaces

ATEsystem_PIRIS

Macros

```
#define PIRIS_ETH_RX_ITER_DELAY 50
```

```
#define PIRIS_ETH_TX_BUF_FUL_WAIT 100
```

Macro Definition Documentation

◆ PIRIS_ETH_RX_ITER_DELAY

```
#define PIRIS_ETH_RX_ITER_DELAY 50
```

◆ PIRIS_ETH_TX_BUF_FUL_WAIT

```
#define PIRIS_ETH_TX_BUF_FUL_WAIT 100
```

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver-Shared

include

[Classes](#) | [Namespaces](#) | [Macros](#)

ATEsystem.PIRIS_serial.h File Reference

```
#include <string> #include <vector>
#include "serial.h"
#include "ATEsystem.PIRIS.h"
```

[Go to the source code of this file.](#)

Classes

class **ATEsystem_PIRIS::CPirisSerial**

Namespaces

ATEsystem_PIRIS

Macros

```
#define PIRIS_SERIAL_BAUDRATE (uint32_t)9600
```

```
#define PIRIS_SERIAL_BITS serial::eightbits
```

```
#define PIRIS_SERIAL_PARITY serial::parity_none
```

```
#define PIRIS_SERIAL_STOPBITS serial::stopbits_one
```

```
#define PIRIS_SERIAL_FLOW serial::flowcontrol_none
```

```
#define PIRIS_SERIAL_MAX_RX 100
```

```
#define PIRIS_SERIAL_EOL "\r\n"
```

Macro Definition Documentation

◆ PIRIS_SERIAL_BAUDRATE

```
#define PIRIS_SERIAL_BAUDRATE (uint32_t)9600
```

◆ PIRIS_SERIAL_BITS

```
#define PIRIS_SERIAL_BITS serial::eightbits
```

◆ PIRIS_SERIAL_EOL

```
#define PIRIS_SERIAL_EOL  "\r\n"
```

◆ PIRIS_SERIAL_FLOW

```
#define PIRIS_SERIAL_FLOW serial::flowcontrol_none
```

◆ PIRIS_SERIAL_MAX_RX

```
#define PIRIS_SERIAL_MAX_RX 100
```

◆ PIRIS_SERIAL_PARITY

```
#define PIRIS_SERIAL_PARITY serial::parity_none
```

◆ PIRIS_SERIAL_STOPBITS

```
#define PIRIS_SERIAL_STOPBITS serial::stopbits_one
```

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver-Shared

src

src Directory Reference

Files

file [**ATEsystem.PIRIS.cpp**](#)

file [**ATEsystem.PIRIS_ethernet.cpp**](#)

file [**ATEsystem.PIRIS_serial.cpp**](#)

Generated by [**doxygen**](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver-Shared

src

Namespaces | Functions

ATEsystem.PIRIS.cpp File Reference

```
#include "pylon/PylonIncludes.h" #include "ATEsystem.PIRIS.h"
#include "ATEsystem.PIRIS_ethernet.h"
#include "ATEsystem.PIRIS_serial.h"
```

Namespaces

ATEsystem_PIRIS

Functions

```
std::ostream & ATEsystem_PIRIS::operator<< (std::ostream &os,  
const Status stat)
```

```
std::ostream & ATEsystem_PIRIS::operator<< (std::ostream &os,  
const PirisDeviceType type)
```

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver-Shared

src

Namespaces

ATEsystem.PIRIS_ethernet.cpp File Reference

```
#include <iostream> #include "pylon/PylonIncludes.h"
#include "ATEsystem.PIRIS.h"
#include "ATEsystem.PIRIS_ethernet.h"
```

Namespaces

ATEsystem_PIRIS

Generated by **doxygen** 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver-Shared

src

Namespaces

ATEsystem.PIRIS_serial.cpp File Reference

```
#include <iostream> #include <iomanip>
#include "serial.h"
#include "ATEsystem.PIRIS.h"
#include "ATEsystem.PIRIS_serial.h"
```

Namespaces

ATEsystem_PIRIS

Generated by **doxygen** 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver-Shared

Macros | Functions | Variables

example.cpp File Reference

```
#include <cstdlib> #include <iostream>
#include <algorithm>
#include "ATEsystem.PIRIS.h"
```

Macros

```
#define FLAG_NO_PYLON_GUI "--no-gui"
```

Functions

```
int main_ethernet (int d)
```

```
int main_serial (int d)
```

```
void check_stat (ATEsystem_PIRIS::Status  
stat)
```

```
void check_err  
(ATEsystem_PIRIS::StatusEx stat)
```

```
void prompt (std::string msg="", std::string  
msg2="")
```

```
uint16_t input_num (int init, int lo, int hi,  
std::string msg)
```

```
bool arg_exists (char **begin, char **end,  
const std::string &option)
```

```
ATEsystem_PIRIS::Status action_wait_handler  
(ATEsystem_PIRIS::IPiris &piris)
```

```
bool input_values (int &f, int &z, int &p, int  
&ir, ATEsystem_PIRIS::DataParams  
&data_param)
```

```
int main (int argc, char *argv[])
```

Variables

```
int8_t status = -20
```

Macro Definition Documentation

◆ FLAG_NO_PYLON_GUI

```
#define FLAG_NO_PYLON_GUI  "--no-gui"
```

Function Documentation

◆ action_wait_handler()

ATEsystem_PIRIS::Status

action_wait_handler

(ATEsystem_PIRIS::IPiris & piris)

◆ arg_exists()

```
bool arg_exists ( char ** begin,  
                  char ** end,  
                  const std::string & option  
                )
```

◆ check_err()

```
void check_err( ATEsystem_PIRIS::StatusEx stat )
```

◆ check_stat()

```
void check_stat( ATEsystem_PIRIS::Status stat )
```

◆ input_num()

```
uint16_t input_num( int      init,  
                    int      lo,  
                    int      hi,  
                    std::string msg  
    )
```

◆ input_values()

```
bool input_values ( int & f,  
                    int & z,  
                    int & p,  
                    int & ir,  
                    ATEsystem_PIRIS::DataParams & data_param  
    )
```

◆ main()

```
int main ( int    argc,  
          char * argv[]  
        )
```

◆ main_etherne()

```
int main_etherne( int d )
```

◆ main_serial()

```
int main_serial ( int d )
```

◆ prompt()

```
void prompt ( std::string msg = "",  
              std::string msg2 = ""  
            )
```

Variable Documentation

◆ status

```
int8_t status = -20
```

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver-Shared

Macros | Functions | Variables

example_etherne.cpp File Reference

```
#include <iostream> #include <cstdlib>
#include <ctime>
#include <mutex>
#include <thread>
#include <fstream>
#include "pylon/PylonIncludes.h"
#include "ATEsystem.PIRIS.h"
```

Macros

```
#define CAMERA_CONFIGURATION "acA2040-35gmATE.pfs"
```

```
#define DEMO_IS_SEQUENCE(x) (x & 0x1)
```

```
#define PYLON_GUI_ENABLED(x) (x & 0x2)
```

Functions

```
int main_etherne
```

```
void check_stat (ATEsystem_PIRIS::Status  
stat)
```

```
void check_err  
(ATEsystem_PIRIS::StatusEx stat)
```

```
void prompt (std::string msg="", std::string  
msg2="")
```

```
uint16_t input_num (int init, int lo, int hi,  
std::string msg)
```

```
ATEsystem_PIRIS::Status action_wait_handler  
(ATEsystem_PIRIS::IPiris &piris)
```

```
bool input_values (int &f, int &z, int &p, int  
&ir, ATEsystem_PIRIS::DataParams  
&data_param)
```

Variables

```
std::mutex mtx
```

```
bool flag_t_run = true
```

Macro Definition Documentation

◆ CAMERA_CONFIGURATION

```
#define CAMERA_CONFIGURATION "acA2040-35gmATE.pfs"
```

◆ DEMO_IS_SEQUENCE

```
#define DEMO_IS_SEQUENCE( x ) (x & 0x1)
```

◆ PYLON_GUI_ENABLED

```
#define PYLON_GUI_ENABLED( x) (x & 0x2)
```

Function Documentation

◆ action_wait_handler()

ATEsystem_PIRIS::Status

action_wait_handler

(ATEsystem_PIRIS::IPiris & piris)

◆ check_err()

```
void check_err( ATEsystem_PIRIS::StatusEx stat )
```

◆ check_stat()

```
void check_stat( ATEsystem_PIRIS::Status stat )
```

◆ input_num()

```
uint16_t input_num( int      init,  
                    int      lo,  
                    int      hi,  
                    std::string msg  
    )
```

◆ input_values()

```
bool input_values ( int & f,  
                    int & z,  
                    int & p,  
                    int & ir,  
                    ATEsystem_PIRIS::DataParams & data_param  
    )
```

◆ main_etherne()

```
int main_etherne( int d )
```

◆ prompt()

```
void prompt ( std::string msg = "",  
              std::string msg2 = ""  
            )
```

Variable Documentation

◆ flag_t_run

```
bool flag_t_run = true
```

◆ mtx

std::mutex mtx

Generated by [doxygen](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver-Shared

Macros | Functions

example_serial.cpp File Reference

```
#include <iostream> #include <cstdlib>
#include <ctime>
#include "serial.h"
#include "ATEsystem.PIRIS.h"
```

Macros

```
#define DEMO_IS_SEQUENCE(x) (x & 0x1)
```

Functions

```
int main_serial (int d)
```

```
void check_stat (ATEsystem_PIRIS::Status  
stat)
```

```
void check_err  
(ATEsystem_PIRIS::StatusEx stat)
```

```
void prompt (std::string msg="", std::string  
msg2="")
```

```
uint16_t input_num (int init, int lo, int hi,  
std::string msg)
```

```
ATEsystem_PIRIS::Status action_wait_handler  
(ATEsystem_PIRIS::IPiris &piris)
```

```
bool input_values (int &f, int &z, int &p, int  
&ir, ATEsystem_PIRIS::DataParams  
&data_param)
```

Macro Definition Documentation

◆ DEMO_IS_SEQUENCE

```
#define DEMO_IS_SEQUENCE( x ) (x & 0x1)
```

Function Documentation

◆ action_wait_handler()

ATEsystem_PIRIS::Status

action_wait_handler

(ATEsystem_PIRIS::IPiris & piris)

◆ check_err()

```
void check_err( ATEsystem_PIRIS::StatusEx stat )
```

◆ check_stat()

```
void check_stat( ATEsystem_PIRIS::Status stat )
```

◆ input_num()

```
uint16_t input_num( int      init,  
                    int      lo,  
                    int      hi,  
                    std::string msg  
    )
```

◆ input_values()

```
bool input_values ( int & f,  
                    int & z,  
                    int & p,  
                    int & ir,  
                    ATEsystem_PIRIS::DataParams & data_param  
    )
```

◆ main_serial()

```
int main_serial ( int d )
```

◆ prompt()

```
void prompt ( std::string msg = "",  
              std::string msg2 = ""  
            )
```

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver

doxygen_doc

doxygen_doc Directory Reference

Generated by [doxygen](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

Here is a list of all file members with links to the files they belong to:

- a -

- action_wait_handler() : [example.cpp](#) , [example_etherne](#)t.cpp , [example_serial.cpp](#)
- arg_exists() : [example.cpp](#)

- c -

- CAMERA_CONFIGURATION : [example_etherne](#)t.cpp
- check_err() : [example.cpp](#) , [example_etherne](#)t.cpp , [example_serial.cpp](#)
- check_stat() : [example.cpp](#) , [example_etherne](#)t.cpp , [example_serial.cpp](#)
- CMD_HOMING : [ATEsystem.PIRIS.h](#)
- CMD_READ_ID : [ATEsystem.PIRIS.h](#)
- CMD_READ_POS : [ATEsystem.PIRIS.h](#)
- CMD_READ_STATE : [ATEsystem.PIRIS.h](#)
- CMD_READ_TYPE : [ATEsystem.PIRIS.h](#)
- CMD_RESET : [ATEsystem.PIRIS.h](#)
- CMD_SET_ABS : [ATEsystem.PIRIS.h](#)
- CMD_SET_REL : [ATEsystem.PIRIS.h](#)
- CMD_SUFFIX : [ATEsystem.PIRIS.h](#)

- d -

- DELIM : [ATEsystem.PIRIS.h](#)
- DEMO_IS_SEQUENCE : [example_etherne](#)t.cpp , [example_serial.cpp](#)

- e -

- ERR0 : **ATEsystem.PIRIS.h**
- ERR1 : **ATEsystem.PIRIS.h**
- ERR2 : **ATEsystem.PIRIS.h**
- ERR3 : **ATEsystem.PIRIS.h**
- ERR4 : **ATEsystem.PIRIS.h**
- ERR5 : **ATEsystem.PIRIS.h**
- ERR6 : **ATEsystem.PIRIS.h**
- ERR7 : **ATEsystem.PIRIS.h**
- ERR_PA : **ATEsystem.PIRIS.h**
- ERR_UN : **ATEsystem.PIRIS.h**

- f -

- FLAG_NO_PYLON_GUI : **example.cpp**
- flag_t_run : **example_ethernet.cpp**
- FW_MIN_STAT_POLL : **ATEsystem.PIRIS.h**

- g -

- GET_VAR_NAME : **ATEsystem.PIRIS.h**

- i -

- input_num() : **example.cpp** , **example_ethernet.cpp** , **example_serial.cpp**
- input_values() : **example.cpp** , **example_ethernet.cpp** , **example_serial.cpp**

- l -

- LEN_CMD_HOMING : **ATEsystem.PIRIS.h**
- LEN_CMD_READ_ID : **ATEsystem.PIRIS.h**
- LEN_CMD_READ_POS : **ATEsystem.PIRIS.h**
- LEN_CMD_READ_STATE1 : **ATEsystem.PIRIS.h**
- LEN_CMD_READ_STATE2 : **ATEsystem.PIRIS.h**
- LEN_CMD_READ_TYPE : **ATEsystem.PIRIS.h**
- LEN_CMD_RESET : **ATEsystem.PIRIS.h**
- LEN_CMD_SET_ABS : **ATEsystem.PIRIS.h**
- LEN_CMD_SET_REL : **ATEsystem.PIRIS.h**

- m -

- main() : `example.cpp`
- main_ethernet() : `example.cpp` , `example_ethernet.cpp`
- main_serial() : `example.cpp` , `example_serial.cpp`
- MSG : `ATEsystem.PIRIS.h`
- mtx : `example_ethernet.cpp`

- o -

- OK : `ATEsystem.PIRIS.h`

- p -

- PIRIS_ETH_RX_ITER_DELAY : `ATEsystem.PIRIS_ethernet.h`
- PIRIS_ETH_TX_BUF_FUL_WAIT : `ATEsystem.PIRIS_ethernet.h`
- PIRIS_SERIAL_BAUDRATE : `ATEsystem.PIRIS_serial.h`
- PIRIS_SERIAL_BITS : `ATEsystem.PIRIS_serial.h`
- PIRIS_SERIAL_EOL : `ATEsystem.PIRIS_serial.h`
- PIRIS_SERIAL_FLOW : `ATEsystem.PIRIS_serial.h`
- PIRIS_SERIAL_MAX_RX : `ATEsystem.PIRIS_serial.h`
- PIRIS_SERIAL_PARITY : `ATEsystem.PIRIS_serial.h`
- PIRIS_SERIAL_STOPBITS : `ATEsystem.PIRIS_serial.h`
- prompt() : `example.cpp` , `example_ethernet.cpp` ,
`example_serial.cpp`
- PYLON_GUI_ENABLED : `example_ethernet.cpp`

- r -

- RESP_FOCUS : `ATEsystem.PIRIS.h`
- RESP_IRIS : `ATEsystem.PIRIS.h`
- RESP_ZOOM : `ATEsystem.PIRIS.h`
- RX_TIMEOUT_NEW_FW : `ATEsystem.PIRIS.h`
- RX_TIMEOUT_OLD_FW : `ATEsystem.PIRIS.h`

- s -

- status : `example.cpp`
- STR_MAX_SIZE : `ATEsystem.PIRIS.h`

- t -

- TRUE_VAL : [ATEsystem.PIRIS.h](#)
- TXT_ALREADY_OPEN : [ATEsystem.PIRIS.h](#)
- TXT_BUSY : [ATEsystem.PIRIS.h](#)
- TXT_CLOSE_FAILED : [ATEsystem.PIRIS.h](#)
- TXT_DEV_SET_FAILED : [ATEsystem.PIRIS.h](#)
- TXT_DEVICE_IS_BUSY : [ATEsystem.PIRIS.h](#)
- TXT_DEVICE_IS_CLOSED : [ATEsystem.PIRIS.h](#)
- TXT_DEVICE_NOT_EXIST : [ATEsystem.PIRIS.h](#)
- TXT_ETHERNET_CAMERA : [ATEsystem.PIRIS.h](#)
- TXT_ETHERNET_DEVICE : [ATEsystem.PIRIS.h](#)
- TXT_ETHERNET_PYLON : [ATEsystem.PIRIS.h](#)
- TXT_FAILED : [ATEsystem.PIRIS.h](#)
- TXT_FOCUS : [ATEsystem.PIRIS.h](#)
- TXT_GENERAL_ERROR : [ATEsystem.PIRIS.h](#)
- TXT_INTERNAL_ERROR : [ATEsystem.PIRIS.h](#)
- TXT_INVALID_DEVICE : [ATEsystem.PIRIS.h](#)
- TXT_IR_FILTER : [ATEsystem.PIRIS.h](#)
- TXT_IR_PRESENT : [ATEsystem.PIRIS.h](#)
- TXT_IRIS : [ATEsystem.PIRIS.h](#)
- TXT_LENS : [ATEsystem.PIRIS.h](#)
- TXT_NAME : [ATEsystem.PIRIS.h](#)
- TXT_OPEN_FAILED : [ATEsystem.PIRIS.h](#)
- TXT_PARSE_ANSWER_FAIL : [ATEsystem.PIRIS.h](#)
- TXT_READ_ERROR : [ATEsystem.PIRIS.h](#)
- TXT_RX_TIMEOUT : [ATEsystem.PIRIS.h](#)
- TXT_SCAN_FAILED : [ATEsystem.PIRIS.h](#)
- TXT_SENS_PRESENT : [ATEsystem.PIRIS.h](#)
- TXT_SERIAL : [ATEsystem.PIRIS.h](#)
- TXT_SUCCESS : [ATEsystem.PIRIS.h](#)
- TXT_SUCCESS2 : [ATEsystem.PIRIS.h](#)
- TXT_TYPE : [ATEsystem.PIRIS.h](#)
- TXT_VERSION : [ATEsystem.PIRIS.h](#)
- TXT_WRITE_ERROR : [ATEsystem.PIRIS.h](#)
- TXT_WRONG_MODE : [ATEsystem.PIRIS.h](#)
- TXT_ZOOM : [ATEsystem.PIRIS.h](#)

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

- `action_wait_handler()` : `example.cpp` , `example_etherne`t.`cpp` ,
`example_serial.cpp`
- `arg_exists()` : `example.cpp`
- `check_err()` : `example.cpp` , `example_etherne`t.`cpp` ,
`example_serial.cpp`
- `check_stat()` : `example.cpp` , `example_etherne`t.`cpp` ,
`example_serial.cpp`
- `input_num()` : `example.cpp` , `example_etherne`t.`cpp` ,
`example_serial.cpp`
- `input_values()` : `example.cpp` , `example_etherne`t.`cpp` ,
`example_serial.cpp`
- `main()` : `example.cpp`
- `main_etherne`t() `: example.cpp` , `example_etherne`t.`cpp`
- `main_serial()` : `example.cpp` , `example_serial.cpp`
- `prompt()` : `example.cpp` , `example_serial.cpp` ,
`example_etherne`t.`cpp`

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

- flag_t_run : [example_ethernet.cpp](#)
- mtx : [example_ethernet.cpp](#)
- status : [example.cpp](#)

Generated by [doxygen](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1
ATEsystem.PIRIS-driver

- c -

- CAMERA_CONFIGURATION : [example_ethernet.cpp](#)
- CMD_HOMING : [ATEsystem.PIRIS.h](#)
- CMD_READ_ID : [ATEsystem.PIRIS.h](#)
- CMD_READ_POS : [ATEsystem.PIRIS.h](#)
- CMD_READ_STATE : [ATEsystem.PIRIS.h](#)
- CMD_READ_TYPE : [ATEsystem.PIRIS.h](#)
- CMD_RESET : [ATEsystem.PIRIS.h](#)
- CMD_SET_ABS : [ATEsystem.PIRIS.h](#)
- CMD_SET_REL : [ATEsystem.PIRIS.h](#)
- CMD_SUFFIX : [ATEsystem.PIRIS.h](#)

- d -

- DELIM : [ATEsystem.PIRIS.h](#)
- DEMO_IS_SEQUENCE : [example_ethernet.cpp](#) ,
[example_serial.cpp](#)

- e -

- ERR0 : [ATEsystem.PIRIS.h](#)
- ERR1 : [ATEsystem.PIRIS.h](#)
- ERR2 : [ATEsystem.PIRIS.h](#)
- ERR3 : [ATEsystem.PIRIS.h](#)
- ERR4 : [ATEsystem.PIRIS.h](#)
- ERR5 : [ATEsystem.PIRIS.h](#)
- ERR6 : [ATEsystem.PIRIS.h](#)
- ERR7 : [ATEsystem.PIRIS.h](#)
- ERR_PA : [ATEsystem.PIRIS.h](#)
- ERR_UN : [ATEsystem.PIRIS.h](#)

- f -

- FLAG_NO_PYLON_GUI : [example.cpp](#)
- FW_MIN_STAT_POLL : [ATEsystem.PIRIS.h](#)

- g -

- GET_VAR_NAME : [ATEsystem.PIRIS.h](#)

- l -

- LEN_CMD_HOMING : [ATEsystem.PIRIS.h](#)
- LEN_CMD_READ_ID : [ATEsystem.PIRIS.h](#)
- LEN_CMD_READ_POS : [ATEsystem.PIRIS.h](#)
- LEN_CMD_READ_STATE1 : [ATEsystem.PIRIS.h](#)
- LEN_CMD_READ_STATE2 : [ATEsystem.PIRIS.h](#)
- LEN_CMD_READ_TYPE : [ATEsystem.PIRIS.h](#)
- LEN_CMD_RESET : [ATEsystem.PIRIS.h](#)
- LEN_CMD_SET_ABS : [ATEsystem.PIRIS.h](#)
- LEN_CMD_SET_REL : [ATEsystem.PIRIS.h](#)

- m -

- MSG : [ATEsystem.PIRIS.h](#)

- o -

- OK : [ATEsystem.PIRIS.h](#)

- p -

- PIRIS_ETH_RX_ITER_DELAY : [ATEsystem.PIRIS_ethernet.h](#)
- PIRIS_ETH_TX_BUF_FUL_WAIT : [ATEsystem.PIRIS_ethernet.h](#)
- PIRIS_SERIAL_BAUDRATE : [ATEsystem.PIRIS_serial.h](#)
- PIRIS_SERIAL_BITS : [ATEsystem.PIRIS_serial.h](#)
- PIRIS_SERIAL_EOL : [ATEsystem.PIRIS_serial.h](#)
- PIRIS_SERIAL_FLOW : [ATEsystem.PIRIS_serial.h](#)
- PIRIS_SERIAL_MAX_RX : [ATEsystem.PIRIS_serial.h](#)
- PIRIS_SERIAL_PARITY : [ATEsystem.PIRIS_serial.h](#)

- PIRIS_SERIAL_STOPBITS : [ATEsystem.PIRIS_serial.h](#)
- PYLON_GUI_ENABLED : [example_ethernet.cpp](#)

- r -

- RESP_FOCUS : [ATEsystem.PIRIS.h](#)
- RESP_IRIS : [ATEsystem.PIRIS.h](#)
- RESP_ZOOM : [ATEsystem.PIRIS.h](#)
- RX_TIMEOUT_NEW_FW : [ATEsystem.PIRIS.h](#)
- RX_TIMEOUT_OLD_FW : [ATEsystem.PIRIS.h](#)

- s -

- STR_MAX_SIZE : [ATEsystem.PIRIS.h](#)

- t -

- TRUE_VAL : [ATEsystem.PIRIS.h](#)
- TXT_ALREADY_OPEN : [ATEsystem.PIRIS.h](#)
- TXT_BUSY : [ATEsystem.PIRIS.h](#)
- TXT_CLOSE_FAILED : [ATEsystem.PIRIS.h](#)
- TXT_DEV_SET_FAILED : [ATEsystem.PIRIS.h](#)
- TXT_DEVICE_IS_BUSY : [ATEsystem.PIRIS.h](#)
- TXT_DEVICE_IS_CLOSED : [ATEsystem.PIRIS.h](#)
- TXT_DEVICE_NOT_EXIST : [ATEsystem.PIRIS.h](#)
- TXT_ETHERNET_CAMERA : [ATEsystem.PIRIS.h](#)
- TXT_ETHERNET_DEVICE : [ATEsystem.PIRIS.h](#)
- TXT_ETHERNET_PYTHON : [ATEsystem.PIRIS.h](#)
- TXT_FAILED : [ATEsystem.PIRIS.h](#)
- TXT_FOCUS : [ATEsystem.PIRIS.h](#)
- TXT_GENERAL_ERROR : [ATEsystem.PIRIS.h](#)
- TXT_INTERNAL_ERROR : [ATEsystem.PIRIS.h](#)
- TXT_INVALID_DEVICE : [ATEsystem.PIRIS.h](#)
- TXT_IR_FILTER : [ATEsystem.PIRIS.h](#)
- TXT_IR_PRESENT : [ATEsystem.PIRIS.h](#)
- TXT_IRIS : [ATEsystem.PIRIS.h](#)
- TXT_LENS : [ATEsystem.PIRIS.h](#)
- TXT_NAME : [ATEsystem.PIRIS.h](#)
- TXT_OPEN_FAILED : [ATEsystem.PIRIS.h](#)
- TXT_PARSE_ANSWER_FAIL : [ATEsystem.PIRIS.h](#)

- `TXT_READ_ERROR` : `ATEsystem.PIRIS.h`
 - `TXT_RX_TIMEOUT` : `ATEsystem.PIRIS.h`
 - `TXT_SCAN_FAILED` : `ATEsystem.PIRIS.h`
 - `TXT_SENS_PRESENT` : `ATEsystem.PIRIS.h`
 - `TXT_SERIAL` : `ATEsystem.PIRIS.h`
 - `TXT_SUCCESS` : `ATEsystem.PIRIS.h`
 - `TXT_SUCCESS2` : `ATEsystem.PIRIS.h`
 - `TXT_TYPE` : `ATEsystem.PIRIS.h`
 - `TXT_VERSION` : `ATEsystem.PIRIS.h`
 - `TXT_WRITE_ERROR` : `ATEsystem.PIRIS.h`
 - `TXT_WRONG_MODE` : `ATEsystem.PIRIS.h`
 - `TXT_ZOOM` : `ATEsystem.PIRIS.h`
-

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATESystem_PIRIS

CPirisEthernet

ATEsystem_PIRIS::CPirisEthernet Member List

This is the complete list of members for **ATEsystem_PIRIS::CPirisEthernet**, including all inherited members.

close(bool ignore_err=false)

Close()

CPirisEthernet()

CPirisMain()

create()

DevHoming()

DevReset()

flush(bool ignore_err=false)

GetFwPollSupport()

GetVerboseLevel()

open(const PirisDevice &dev, VerboseLevel verbose=VerboseLevel::NC)

Open(void *dev, PirisDeviceType type, VerboseLevel verbose=VerboseL

Open(const PirisDevice &dev, VerboseLevel verbose=VerboseLevel::NC)

read(std::string &data)

ReadID()

ReadParams()

ReadPosition()

ReadState()

remove()

rx_timeout

```
scan(std::vector< Pylon::CDeviceInfo > &devices, bool verbose=false)
ScanEthernet(std::vector< Pylon::CDeviceInfo > &devices, bool verbose=false)
ScanSerial(std::vector< serial::PortInfo > &devices, bool verbose=false)
SetAbsolute(uint16_t focus=0, uint16_t zoom=0, uint16_t iris=0, bool ir_filter=false)
SetAbsolute(const FocusZoomIris< uint16_t > &values, bool ir_filter=false)
SetRelative(int16_t focus=0, int16_t zoom=0, int16_t iris=0)
SetRelative(const FocusZoomIris< int16_t > &values)
SetVerboseLevel(VerboseLevel level)
state_get()
state_is(State _state)
state_reset()
state_set(State _state)
verbose_is(VerboseLevel _verbose)
verbose_set(VerboseLevel _verbose)
write(const std::string data)
~CPirisEthernet()
~CPirisMain()
```

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver > ATEsystem.PIRIS-driver-Shared > include >

ATEsystem.PIRIS_ethernet.h

Go to the documentation of this file.

```
1  ****
2  *
3  * Author:      Jakub Perez
4  * File:        ATEsystem.PIRIS_ethernet.h
5  * Project:     ATEsystem.PIRIS-driver
6  * Version:    1.0.1
7  * Company:    ATEsystem s.r.o
8  * Date:       2018/11/23 12:30
9  * License:    WTFPL
10 * Require:   C++17, Pylon 5, GenICam
11 *
12 ****
13
14 #pragma once
15
16 #include <string>
17 #include <vector>
18
19 #include "pylon/PylonIncludes.h"
20
21 #include "ATEsystem.PIRIS.h"
22
23 #define PIRIS_ETH_RX_ITER_DELAY      50
24 #define PIRIS_ETH_TX_BUF_FUL_WAIT 100
25
26
```

```
27 | namespace ATESystem_PIRIS
28 | {
29 |     class CPirisEthernet final : public
30 |         CPirisMain
31 |     {
32 |         public:
33 |
34 |             static int16_t
35 |             scan(std::vector<Pylon::CDeviceInfo>& devices,
36 |                   bool verbose = false);
37 |             static IDevice* create();
38 |
39 |             CPirisEthernet();
40 |             virtual ~CPirisEthernet();
41 |             virtual void remove();
42 |
43 |         protected:
44 |
45 |             virtual Status open(const
46 |             PirisDevice& dev, VerboseLevel verbose =
47 |             VerboseLevel::NONE);
48 |             virtual Status close(bool ignore_err
49 |             = false);
50 |             virtual Status write(const
51 |             std::string data);
52 |             virtual Status read(std::string&
53 |             data);
54 |             virtual Status flush(bool ignore_err
55 |             = false);
56 |
57 |         private:
58 |
59 |             bool cam_overiden;
60 |
61 |             Pylon::IPylonDevice* pDevice;
62 |             Pylon::CInstantCamera* pCamera;
```

```
58     GenApi::CIntegerPtr SerialIOControl;
59     GenApi::CIntegerPtr SerialIOSTatus;
60     GenApi::CIntegerPtr SerialIODataOut;
61     GenApi::CCommandPtr
62         SerialIODataOutExecute;
63     GenApi::CBooleanPtr
64         SerialIOSTatusDataOutBufferFull;
65     GenApi::CBooleanPtr
66         SerialIOSTatusDataOutBufferEmpty;
67     GenApi::CIntegerPtr SerialIODataIn;
68     GenApi::CCommandPtr
69         SerialIODataInExecute;
70     GenApi::CBooleanPtr
71         SerialIOSTatusDataInBufferFull;
72     GenApi::CBooleanPtr
73         SerialIOSTatusDataInBufferEmpty;
74     Status dev_set_up();
75 }
```

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATESystem_PIRIS

CPirisMain

ATEsystem_PIRIS::CPirisMain Member List

This is the complete list of members for **ATEsystem_PIRIS::CPirisMain**, including all inherited members.

[Close\(\)](#)

[CPirisMain\(\)](#)

[DevHoming\(\)](#)

[DevReset\(\)](#)

[GetFwPollSupport\(\)](#)

[GetVerboseLevel\(\)](#)

[Open\(void *dev, PirisDeviceType type, VerboseLevel verbose=VerboseL](#)

[Open\(const PirisDevice &dev, VerboseLevel verbose=VerboseLevel::NC](#)

[ReadID\(\)](#)

[ReadParams\(\)](#)

[ReadPosition\(\)](#)

[ReadState\(\)](#)

[rx_timeout](#)

[ScanEthernet\(std::vector< Pylon::CDeviceInfo > &devices, bool verbose=false\)](#)

[ScanSerial\(std::vector< serial::PortInfo > &devices, bool verbose=false\)](#)

[SetAbsolute\(uint16_t focus=0, uint16_t zoom=0, uint16_t iris=0, bool ir_filter=false\)](#)

[SetAbsolute\(const FocusZoomIris< uint16_t > &values, bool ir_filter=false\)](#)

[SetRelative\(int16_t focus=0, int16_t zoom=0, int16_t iris=0\)](#)

[SetRelative\(const FocusZoomIris< int16_t > &values\)](#)

[SetVerboseLevel\(VerboseLevel level\)](#)

```
state_get()
state_is(State _state)
state_reset()
state_set(State _state)
verbose_is(VerboseLevel _verbose)
verbose_set(VerboseLevel _verbose)
~CPirisMain()
```

Generated by [doxygen](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver > ATEsystem.PIRIS-driver-Shared > include >

ATEsystem.PIRIS.h

Go to the documentation of this file.

```
1  ****
2  *
3  * Author:      Jakub Perez
4  * File:        ATEsystem.PIRIS.h
5  * Project:     ATEsystem.PIRIS-driver
6  * Version:    1.0.1
7  * Company:    ATEsystem s.r.o
8  * Date:       2018/11/23 09:29
9  * License:    WTFPL
10 * Require:   C++17, Pylon 5, GenICam, Serial
11 *
12 ****
13
14 #pragma once
15
16 #ifdef _WIN32
17     #include <windows.h>
18
19     #define Interface    class //__interface
20     #define _sscanf        sscanf_s
21     #define _Sleep(ms)    Sleep(ms)
22
23 #elif __linux__
24     #include <unistd.h>
25     #include <stdlib.h>
26     #include <cstdint>
```

```
27
28     #define Interface    struct
29     #define _sscanf        sscanf
30     #define _Sleep(ms)    usleep(ms * 1000)
31
32 #else
33     #error Unknown / unsupported platform
34 #endif
35
36 #include <string>
37 #include <map>
38 #include <vector>
39 #include <list>
40
41 #include "pylon/PylonIncludes.h"
42 #include "serial.h"
43
44
45 // ERROR TEXT
46 #define OK                  "OK"
47 #define ERR0                "Command was not
recognized."
48 #define ERR1                "Wrong input
parameter."
49 #define ERR2                "Device is not
initialized."
50 #define ERR3                "Internal driver
circuit is in error state (overheat,
undervoltage)."
51 #define ERR4                "Wrong lens type."
52 #define ERR5                "Damaged or missing
rear sensor."
53 #define ERR6                "Step generator timer
overflow."
54 #define ERR7                "Device is busy."
55 #define ERR_UN              "Unknown error."
56 #define ERR_PA              "Parse answer failed."
```

```

57 |
58 // STATUS TEXT
59 #define TXT_SUCCESS "Success"
60 #define TXT_GENERAL_ERROR "General Error"
61 #define TXT_OPEN_FAILED "Open failed"
62 #define TXT_CLOSE_FAILED "Close failed"
63 #define TXT_ALREADY_OPEN "Already open"
64 #define TXT_DEVICE_IS_BUSY "Device is busy"
65 #define TXT_DEVICE_IS_CLOSED "Device is closed"
66 #define TXT_INVALID_DEVICE "Invalid device"
67 #define TXT_WRITE_ERROR "Write error"
68 #define TXT_READ_ERROR "Read error"
69 #define TXT_SCAN_FAILED "Scan failed"
70 #define TXT_INTERNAL_ERROR "Internal error"
71 #define TXT_RX_TIMEOUT "Receive timeout"
72 #define TXT_DEV_SET_FAILED "Device set failed"
73 #define TXT_WRONG_MODE "Wrong mode"
74 #define TXT_DEVICE_NOT_EXIST "Device does not exist"
75 #define TXT_PARSE_ANSWER_FAIL "Parse answer failed"
76 |
77 // DEVICE MODE TEXT
78 #define TXT_ETHERNET_DEVICE "Ethernet device"
79 #define TXT_ETHERNET_PYLON "Ethernet pylon (camera overiden)"
80 #define TXT_ETHERNET_CAMERA "Ethernet camera"

```

```

    camera (camera overiden)"
81 #define TXT_SERIAL           "Serial"
82
83 // VERBOSE OUTPUT TEXT
84 #define TXT_SUCCESS2         "success"
85 #define TXT_FAILED           "failed"
86 #define TXT_NAME              "name"
87 #define TXT_VERSION           "version"
88 #define TXT_FOCUS             "focus"
89 #define TXT_ZOOM              "zoom"
90 #define TXT_IRIS              "iris"
91 #define TXT_LENS              "lens"
92 #define TXT_IR_FILTER          "ir_filter"
93 #define TXT_IR_PRESENT         "ir_present"
94 #define TXT_SENS_PRESENT       "sens_present"
95 #define TXT_BUSY               "busy"
96 #define TXT_TYPE               "type"
97
98 // ATE PIRIS PROTOCOL
99 #define CMD_READ_ID           "IDN"
100 #define CMD_READ_POS           "GP"
101 #define CMD_READ_TYPE          "GT"
102 #define CMD_READ_STATE         "GS"
103 #define CMD_RESET              "RST"
104 #define CMD_HOMING             "INI"
105 #define CMD_SET_ABS            "SETA:"
106 #define CMD_SET_REL            "SETR:"
107 #define CMD_SUFFIX             "\r\n"
108 #define RESP_FOCUS             "F%hu"
109 #define RESP_ZOOM              "Z%hu"
110 #define RESP_IRIS              "P%hu"
111 #define TRUE_VAL               '1'
112 #define DELIM                 ";"
113
114 // RESPONSE CHAR LEN
115 #define LEN_CMD_READ_ID        3
116 #define LEN_CMD_READ_POS        5

```

```
117 #define LEN_CMD_READ_TYPE    7
118 #define LEN_CMD_READ_STATE1   2
119 #define LEN_CMD_READ_STATE2   3
120 #define LEN_CMD_RESET        0
121 #define LEN_CMD_HOMING       1
122 #define LEN_CMD_SET_ABS      1
123 #define LEN_CMD_SET_REL      1
124
125 #define FW_MIN_STAT_POLL     1, 7, 2 // minimum fw version which support cyclic status poll reading
126
127 #define RX_TIMEOUT_OLD_FW    20000
128 #define RX_TIMEOUT_NEW_FW    1000
129
130 #define MSG(x)                x CMD_SUFFIX
131
132 #define STR_MAX_SIZE    200
133 #define GET_VAR_NAME(Variable) (#Variable)
134
135
136 namespace ATEsystem_PIRIS
137 {
138     typedef int16_t Status_t;
139
140     typedef int16_t DevID_t;
141
142     enum Mode : uint8_t
143     {
144         SERIAL = 0,
145         ETHERNET = 1
146     };
147
148     enum Status : int16_t
149     {
150         SUCCESS = 0,
151         GENERAL_ERROR = -1,
```

```
164     OPEN_FAILED = -2,
165     CLOSE_FAILED = -3,
166     ALREADY_OPEN = -4,
167     DEVICE_IS_BUSY = -5,
168     DEVICE_IS_CLOSED = -6,
169     INVALID_DEVICE = -7,
170     WRITE_ERROR = -8,
171     READ_ERROR = -9,
172     SCAN_FAILED = -10,
173     INTERNAL_ERROR = -11,
174     RX_TIMEOUT = -12,
175     DEV_SET_FAILED = -13,
176     WRONG_MODE = -14,
177     DEVICE_NOT_EXIST = -15,
178     PARSE_ANSWER_FAIL = -16,
179     UNDEFINED_ERROR = -20
180 };
181 std::ostream& operator<<(std::ostream&
182     os, const Status stat);
183
184 enum State : int8_t
185 {
186     UNKNOWN = 0,
187     CLOSED = 1,
188     BUSY = 2,
189     READY = 3,
190 };
191
192
193 enum PirisDeviceType : uint8_t
194 {
195     ETHERNET_DEVICE = 0,
196     ETHERNET_PYLON = 1,
197     ETHERNET_CAMERA = 2,
198     SERIAL_UART = 3
199 };
200 std::ostream& operator<<(std::ostream&
201     os, const PirisDeviceType type);
```

```
206
210     enum VerboseLevel : uint8_t
211     {
212         NONE = 0,
213         BASIC = 1,
214         FULL = 2
215     };
216
220     enum YesNoNA : uint8_t
221     {
222         NOT_AVAILABLE = 0,
223         YES = 1,
224         NO = 2
225     };
226
227 /**
***** */
228
232     class Utils
233     {
234     public:
235
236         static void CpyCStr2CStr(char* dest,
237             const char* source, size_t max_len);
238         static void CopySTLStr2CStr(char*
239             dest, std::string source);
240         static std::string
241             CopyCStr2STLStr(const char* src);
242         static std::string PadStr(const int
243             total, const int minus = 0, const char ch =
244             ' ');
245     };
246 /**
***** */
247
```

```
243
248     class PirisDevice
249     {
250         private:
251
252             const Mode mode = Mode::ETHERNET;
253             const PirisDeviceType type =
254                 PirisDeviceType::ETHERNET_DEVICE;
255             Pylon::CDeviceInfo dummy_dev;
256             serial::PortInfo dummy_port;
257
258             Pylon::CDeviceInfo& device_eth;
259             Pylon::IPylonDevice* pylon_dev =
260                 nullptr;
261             Pylon::CInstantCamera* camera =
262                 nullptr;
263             serial::PortInfo& serial;
264
265         public:
266
267             PirisDevice(Pylon::CDeviceInfo&
268                         device);
269             PirisDevice(Pylon::IPylonDevice*
270                         pylon_dev);
271             PirisDevice(Pylon::CInstantCamera*
272                         camera);
273             PirisDevice(serial::PortInfo&
274                         serial);
275
276             Pylon::CDeviceInfo& get_device_eth()
277             const;
278             Pylon::IPylonDevice* get_pylon_dev()
279             const;
280             Pylon::CInstantCamera* get_camera()
281             const;
282             serial::PortInfo& get_serial() const;
283             const Mode get_mode() const;
```

```
274         const PirisDeviceType get_type()
275         const;
276         std::string ToString() const;
277         friend std::ostream&
278         operator<<(std::ostream& strm, const
279         PirisDevice& dev)
280         {
281             strm << dev.ToString();
282             return strm;
283         }
284     /*
***** */
285
286     class ErrorCluster
287     {
288     private:
289
290         bool ok;
291         int8_t err_num;
292         std::string msg;
293
294         static std::map<std::string,
295         std::string> errors;
296
297     public:
298
299         ErrorCluster(std::string err_raw =
300             "", bool overide = false);
301         const bool getOk() const;
302         const int8_t getErrNum() const;
303         const std::string getMsg() const;
304         const std::tuple<bool, int8_t,
305         std::string> getErrorCluster() const;
```

```
306
307         std::string ToString() const;
308         friend std::ostream&
309     operator<<(std::ostream& strm, const
310     ErrorCluster& err)
311     {
312         strm << err.ToString();
313         return strm;
314     }
315 */
***** */
316
320     class Version
321     {
322     private:
323
324         const uint16_t _major;
325         const uint16_t _minor;
326         const uint16_t _revision;
327
328     public:
329
330         Version(uint16_t major = 0, uint16_t
331             minor = 0, uint16_t revision = 0);
331         const uint16_t getMajor() const;
332         const uint16_t getMinor() const;
333         const uint16_t getRevision() const;
334         const std::tuple<uint16_t, uint16_t,
335             uint16_t> getVersion() const;
335
336         std::string ToString() const;
337         friend std::ostream&
338     operator<<(std::ostream& strm, const Version&
339     ver)
```

```
338     {
339         strm << ver.ToString();
340         return strm;
341     }
342     friend bool operator==(const Version&
343     ver1, const Version& ver2)
344     {
345         return (ver1._major ==
346             ver2._major && ver1._minor == ver2._minor &&
347             ver1._revision == ver2._revision);
348     }
349     friend bool operator>(const Version&
350     ver1, const Version& ver2)
351     {
352         return (ver1._major > ver2._major ||
353             (ver1._major ==
354                 ver2._major && ver1._minor > ver2._minor) ||
355             (ver1._major ==
356                 ver2._major && ver1._minor == ver2._minor &&
357                 ver1._revision > ver2._revision));
358     }
359     friend bool operator>=(const Version&
360     ver1, const Version& ver2)
```

```
    ||
361             (ver1._major == ver2._major
&& ver1._minor < ver2._minor) ||
362             (ver1._major == ver2._major
&& ver1._minor == ver2._minor && ver1._revision
< ver2._revision));
363         }
364     friend bool operator<=(const Version&
365         ver1, const Version& ver2)
366     {
367         return (ver1._major <=
368             ver2._major ||
369             (ver1._major == ver2._major
&& ver1._minor <= ver2._minor) ||
370             (ver1._major == ver2._major
&& ver1._minor == ver2._minor && ver1._revision
<= ver2._revision));
371     }
372 }

373 /*
***** */
374
375
376
377 template <class T = uint16_t>
378 class FocusZoomIris
379 {
380     private:
381
382         int pad_len;
383
384         const T _focus;
385         const T _zoom;
386         const T _iris;
387
388     public:
389 }
```

```
390     FocusZoomIris(T focus = NULL, T zoom
= NULL, T iris = NULL, int str_pad_len = 25) :
391         _focus(focus), _zoom(zoom),
392         _iris(iris), pad_len(str_pad_len)
393     {
394     }
395     const T getFocus() const
396     {
397         return this->_focus;
398     }
399     const T getZoom() const
400     {
401         return this->_zoom;
402     }
403     const T getIris() const
404     {
405         return this->_iris;
406     }
407     const std::tuple<T, T, T>
408     getFocusZoomIris() const
409     {
410         return std::make_tuple(this-
411             >_focus, this->_zoom, this->_iris);
412     }
413     std::string ToString() const
414     {
415         std::stringstream ret;
416         ret << TXT_FOCUS << "=" <<
        Utils::PadStr(14, 5) << this->_focus <<
        std::endl
417             << Utils::PadStr(pad_len) <<
        TXT_ZOOM << "=" << Utils::PadStr(14, 4) <<
        this->_zoom << std::endl
418             << Utils::PadStr(pad_len) <<
        TXT_IRIS << "=" << Utils::PadStr(14, 4) <<
        this->_iris;
```

```
417         return ret.str();
418     }
419
420     friend std::ostream&
421     operator<<(std::ostream& strm, const
422     FocusZoomIris& vals)
423     {
424         strm << vals.ToString();
425         return strm;
426     }
427 }
 ****
 ****
428
429 class StatusEx : public ErrorCluster
430 {
431 public:
432
433     const Status status;
434     StatusEx(Status status =
435             Status::GENERAL_ERROR, ErrorCluster err =
436             ErrorCluster());
437
438 }
439
440 class DataID
441 {
442 public:
443
444     const std::string name;
445     const Version version;
446
447     DataID(std::string name = "", Version
448             version = Version());
449
450     DataID();
451 }
```

```
456     class DataPosition
457     {
458     public:
459
460         const FocusZoomIris<uint16_t>
461             position;
462         const bool ir_filter;
463
464     DataPosition(FocusZoomIris<uint16_t>
465         pos = FocusZoomIris<uint16_t>(), bool ir =
466         false);
467     };
468
469     class DataParams
470     {
471     public:
472
473         const FocusZoomIris<uint16_t>
474             max_value;
475         const std::string lens;
476         const bool ir_present;
477         const bool sens_present;
478
479     DataParams(FocusZoomIris<uint16_t>
480         max = FocusZoomIris<uint16_t>(), std::string
481         lens = "",
482         bool ir = false, bool sens
483         = false);
484     };
485
486     class DataState
487     {
488     public:
489
490         const FocusZoomIris<bool> state;
491         const bool motors_busy;
```

```
492     DataState(FocusZoomIris<bool> state =
493         FocusZoomIris<bool>(), bool busy = false);
494     };
495     /*
496      ****
497      ****
498      ****
499      ****
500      Interface IDevice
501      {
502      public:
503
504      //*****
505      * open/close
506      ****
507      **** */
512     virtual Status Open(void* dev,
513         PirisDeviceType type, VerboseLevel verbose =
514             VerboseLevel::NONE) = 0;
513
520     virtual Status Open(const
521         PirisDevice& dev, VerboseLevel verbose =
522             VerboseLevel::NONE) = 0;
521
526     virtual Status Close() = 0;
527
528     //*****
529     *** read
530     ****
531     **** */
534     virtual std::tuple<StatusEx, DataID>
535         ReadID() = 0;
535
540     virtual std::tuple<StatusEx,
541         DataPosition> ReadPosition() = 0;
```

```

541
546         virtual std::tuple<StatusEx,
547             DataParams> ReadParams() = 0;
547
552         virtual std::tuple<StatusEx,
553             DataState> ReadState() = 0;
553
554 //*****
554
554     reset/homing
554 ****
554 */
559     virtual StatusEx DevReset() = 0;
560
565     virtual StatusEx DevHoming() = 0;
566
567 //*****
567
567     **** set
567 ****
567 */
576     virtual StatusEx SetAbsolute(uint16_t focus
576         = 0,
577                                     uint16_t
577         zoom = 0,
578                                     uint16_t
578         iris = 0,
579                                     bool
579         ir_filter = false) = 0;
580
589     virtual StatusEx SetAbsolute(const
589         FocusZoomIris<uint16_t>& values,
590                                     bool
590         ir_filter = false) = 0;
591
600     virtual StatusEx SetRelative(int16_t
600         focus = 0,

```

```
601 |                                     int16_t
602 |     zoom = 0,
603 |     iris = 0) = 0;
604 |
612 |         virtual StatusEx SetRelative(const
613 |             FocusZoomIris<int16_t>& values) = 0;
614 |
615 | //***** utils *****
616 | /**
617 | *  @file
618 | *  @brief Implementation of the IDevice interface.
619 | */
620 |         virtual YesNoNA GetFwPollSupport() =
621 |             0;
622 |
623 |         virtual VerboseLevel
624 |             GetVerboseLevel() = 0;
625 |
626 |         virtual void
627 |             SetVerboseLevel(VerboseLevel level) = 0;
628 |
629 |         virtual ~IDevice() = 0;
630 |
631 |     };
632 |
633 |     Interface IComm
634 |     {
635 |         public:
636 |
637 |             virtual Status open(const
638 |                 PirisDevice& dev, VerboseLevel verbose =
639 |                     VerboseLevel::NONE) = 0;
640 |
641 |             virtual Status close(bool ignore_err
642 |                 = false) = 0;
643 |
644 |     };
645 |
646 |     virtual Status
647 |         write(const char* file, const void* data, size_t size) = 0;
648 |
649 |     virtual Status
650 |         read(void* data, size_t size) = 0;
651 |
652 |     virtual Status
653 |         seek(size_t offset, SeekMode mode) = 0;
654 |
655 |     virtual Status
656 |         tell(size_t* offset) = 0;
657 |
658 |     virtual Status
659 |         flush() = 0;
660 |
661 |     virtual Status
662 |         sync() = 0;
663 |
664 |     virtual Status
665 |         syncfs() = 0;
666 |
667 |     virtual Status
668 |         ioctl(int fd, int request, ...)
```

```
664         virtual Status write(const
665             std::string data) = 0;
666
671         virtual Status read(std::string&
672             data) = 0;
673
678         virtual Status flush(bool ignore_err
679             = false) = 0;
680
683         virtual void remove() = 0;
684
685         virtual ~IComm() = 0;
686     };
687
688 /*
***** */
689
693     class CPirisMain : public IComm, public
694         IDevice
695     {
696     public:
697
698         static int16_t
699             ScanEthernet(std::vector<Pylon::CDeviceInfo>&
700                 devices, bool verbose = false);
701
702         static int16_t
703             ScanSerial(std::vector<serial::PortInfo>&
704                 devices, bool verbose = false);
705
706         Status Open(void* dev,
707             PirisDeviceType type, VerboseLevel verbose =
708             VerboseLevel::NONE);
709
710         Status Open(const PirisDevice& dev,
711             VerboseLevel verbose = VerboseLevel::NONE);
712
713         Status Close();
714
```

```
704         std::tuple<StatusEx, DataID>
705             ReadID();           // IDN - !! need to call
706             this on every start to set timeouts
707         std::tuple<StatusEx, DataPosition>
708             ReadPosition(); // GP
709         std::tuple<StatusEx, DataParams>
710             ReadParams();    // GT
711         std::tuple<StatusEx, DataState>
712             ReadState();     // GS
713
714         StatusEx DevReset();      // RST
715         StatusEx DevHoming();   //INI
716
717         StatusEx SetAbsolute(uint16_t focus =
718             0,
719             uint16_t zoom =
720             0,
721             uint16_t iris =
722             0,
723             bool ir_filter =
724             false);
725
726         StatusEx SetAbsolute(const
727             FocusZoomIris<uint16_t>& values,    //
728             SETA:FX;ZX;PX;IX
729             bool ir_filter =
730             false);
731
732         StatusEx SetRelative(int16_t focus =
733             0,
734             int16_t zoom =
735             0,
736             int16_t iris =
737             0);
738
739         StatusEx SetRelative(const
740             FocusZoomIris<int16_t>& values); // //
741             SETR:FX;ZX;PX
742
743         YesNoNA GetFwPollSupport();
```

```
724         VerboseLevel GetVerboseLevel();
725         void SetVerboseLevel(VerboseLevel
726             level);
727
728     CPirisMain();
729     virtual ~CPirisMain() {};
730
731     protected:
732
733         uint16_t rx_timeout =
734             RX_TIMEOUT_NEW_FW;
735
736         void state_set(State _state);
737         State state_get();
738         bool state_is(State _state);
739         void state_reset();
740         bool verbose_is(VerboseLevel
741             _verbose);
742
743         void verbose_set(VerboseLevel
744             _verbose);
745
746         private:
747
748         VerboseLevel _verbose;
749         State state = State::CLOSED;
750         State _state = State::CLOSED;
751         YesNoNA fw_poll_support =
752             YesNoNA::NOT_AVAILABLE; // device firmware is
753             1.7.2 and higher (stat read poll supp)
754
755         ErrorCluster parse_answer(std::string
756             answer_in, std::vector<std::string>&
757             tokens_out, uint8_t n);
758
759         StatusEx
760         send_msg(std::vector<std::string>& tokens_out,
761             std::string name, std::string cmd, uint8_t
762             len);
```

```
750     };
751
752 /* ****
753 ****
754 ****
755 ****
756 ****
757 template <class T>
758 class SmartPointer
759 {
760     private:
761         T* m_pT;
762
763     public:
764         SmartPointer(void) noexcept:
765             m_pT(NULL)
766         {}
767     }
768
769     ~SmartPointer(void)
770     {
771         delete m_pT;
772     }
773
774     void operator=(T *pB)
775     {
776         m_pT = dynamic_cast<T*>(pB);
777     }
778
779     operator T*(void) const
780     {
781         return m_pT;
782     }
783
784     T& operator*(void) const
785     {
786         return *m_pT;
```

```
787     }
788
789     T& operator()(void) const
790     {
791         return *m_pT;
792     }
793
794     T* operator->(void) const
795     {
796         return m_pT;
797     }
798
799     bool IsValid() const throw()
800     {
801         return m_pT != NULL;
802     }
803
804     operator bool(void) const throw()
805     {
806         return m_pT != NULL;
807     }
808
809     bool operator==(T* pT) const
810     {
811         return m_pT == pT;
812     }
813 };
814
815     typedef SmartPointer<IDevice> IPiris;
816
817     /*
818     ****
819     ****
820     */
821
822     typedef IDevice* (*CreateDeviceFn)(void);
823
824     class Factory
```

```
832     {
833     private:
834
835         typedef std::map<Mode,
836             CreateDeviceFn> FactoryMap;
837         typedef std::map<DevID_t,
838             std::pair<std::string, IDevice*>> DeviceMap;
839         // CPirisMain ?
840         FactoryMap _factoryMap;
841         DeviceMap _deviceMap;
842
843     public:
844
845         Factory(Factory const&) = delete;
846         Factory &operator=(Factory const&) =
847             delete;
848
849         ~Factory() { _factoryMap.clear(); }
850
851         static Factory* getInstance()
852         {
853             static Factory instance;
854             return &instance;
855         }
856
857         DevID_t CreateDevice(Mode mode =
858             Mode::ETHERNET);
859
860         DevID_t CreateDevice(const
861             std::string& name, Mode mode);
862
863         IDevice* GetDeviceInstance(DevID_t
864             id);
865
866         private:
867
868         private:
869
870         private:
```

```
886         std::string GetDeviceName(DevID_t
887             id);
887
893         Status RemoveDevice(DevID_t id);
894     };
895 }
```

Generated by [doxygen](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATESystem_PIRIS

CPirisSerial

ATEsystem_PIRIS::CPirisSerial Member List

This is the complete list of members for **ATEsystem_PIRIS::CPirisSerial**, including all inherited members.

Close()

close(bool ignore_err=false)

CPirisMain()

CPirisSerial()

create()

DevHoming()

DevReset()

flush(bool ignore_err=false)

GetFwPollSupport()

GetVerboseLevel()

Open(void *dev, PirisDeviceType type, VerboseLevel verbose=VerboseL

Open(const PirisDevice &dev, VerboseLevel verbose=VerboseLevel::NC

open(const PirisDevice &dev, VerboseLevel verbose=VerboseLevel::NC

read(std::string &data)

ReadID()

ReadParams()

ReadPosition()

ReadState()

remove()

rx_timeout

```
scan(std::vector< serial::PortInfo > &devices, bool verbose=false)
ScanEthernet(std::vector< Pylon::CDeviceInfo > &devices, bool verbose)
ScanSerial(std::vector< serial::PortInfo > &devices, bool verbose=false)
SetAbsolute(uint16_t focus=0, uint16_t zoom=0, uint16_t iris=0, bool ir_filter=false)
SetAbsolute(const FocusZoomIris< uint16_t > &values, bool ir_filter=false)
SetRelative(int16_t focus=0, int16_t zoom=0, int16_t iris=0)
SetRelative(const FocusZoomIris< int16_t > &values)
SetVerboseLevel(VerboseLevel level)
state_get()
state_is(State _state)
state_reset()
state_set(State _state)
verbose_is(VerboseLevel _verbose)
verbose_set(VerboseLevel _verbose)
write(const std::string data)
~CPirisMain()
~CPirisSerial()
```

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem.PIRIS-driver > ATEsystem.PIRIS-driver-Shared > include >

ATEsystem.PIRIS_serial.h

Go to the documentation of this file.

```
1  ****
2  *
3  * Author:      Jakub Perez
4  * File:        ATEsystem.PIRIS_serial.h
5  * Project:     ATEsystem.PIRIS-driver
6  * Version:    1.0.1
7  * Company:    ATEsystem s.r.o
8  * Date:       2018/12/6 18:30
9  * License:    WTFPL
10 * Require:   C++17, Serial
11 *
12 ****
13
14 #pragma once
15
16 #include <string>
17 #include <vector>
18
19 #include "serial.h"
20
21
22 #include "ATEsystem.PIRIS.h"
23
24 #define PIRIS_SERIAL_BAUDRATE
25 (uint32_t)9600
26 #define PIRIS_SERIAL_BITS
```

```
    serial::eightbits
26| #define PIRIS_SERIAL_PARITY
    serial::parity_none
27| #define PIRIS_SERIAL_STOPBITS
    serial::stopbits_one
28| #define PIRIS_SERIAL_FLOW
    serial::flowcontrol_none
29|
30| #define PIRIS_SERIAL_MAX_RX      100
31| #define PIRIS_SERIAL_EOL        "\r\n"
32|
33|
34| namespace ATESystem_PIRIS
35| {
39|     class CPirisSerial final : public
    CPirisMain
40|     {
41|         public:
42|
43|             static int16_t
    scan(std::vector<serial::PortInfo>& devices,
    bool verbose = false);
44|             static IDevice* create();
45|
46|             CPirisSerial();
47|             virtual ~CPirisSerial();
48|             virtual void remove();
49|
50|         protected:
51|
52|             virtual Status open(const
    PirisDevice& dev, VerboseLevel verbose =
    VerboseLevel::NONE);
53|             virtual Status close(bool ignore_err
    = false);
54|             virtual Status write(const
    std::string data);
```

```
55 |         virtual Status read(std::string&
56 |             data);
57 |         virtual Status flush(bool ignore_err
58 |             = false);
59 |
60 |     private:
61 |
62 |         serial::Serial* serial;
63 |
64 |     };
65 }
```

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS > DataID

ATEsystem_PIRIS::DataID Member List

This is the complete list of members for **ATEsystem_PIRIS::DataID**, including all inherited members.

DataID(std::string name="", Version version=Version())	ATEsystem_PIF
name	ATEsystem_PIF
version	ATEsystem_PIF

Generated by **doxygen** 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS

DataParams

ATEsystem_PIRIS::DataParams Member List

This is the complete list of members for **ATEsystem_PIRIS::DataParams**, including all inherited members.

DataParams(FocusZoomIris< uint16_t > max=FocusZoomIris< uint16_t >
ir_present
lens
max_value
sens_present

Generated by **doxygen** 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS > DataPosition >

ATEsystem_PIRIS::DataPosition Member List

This is the complete list of members for **ATEsystem_PIRIS::DataPosition**, including all inherited members.

**DataPosition(FocusZoomIris< uint16_t > pos=FocusZoomIris< uint16_t ir_filter
position**

Generated by doxygen 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS

DataState

ATEsystem_PIRIS::DataState Member List

This is the complete list of members for **ATEsystem_PIRIS::DataState**, including all inherited members.

DataState(FocusZoomIris< bool > state=FocusZoomIris< bool >(), bool
motors_busy
state

Generated by **doxygen** 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATESystem_PIRIS

ErrorCluster

ATEsystem_PIRIS::ErrorCluster Member List

This is the complete list of members for **ATEsystem_PIRIS::ErrorCluster**, including all inherited members.

<code>ErrorCluster(std::string err_raw="", bool override=false)</code>	ATEsystem_F
<code>getErrNum() const</code>	ATEsystem_F
<code>getErrorCluster() const</code>	ATEsystem_F
<code>getMsg() const</code>	ATEsystem_F
<code>getOk() const</code>	ATEsystem_F
<code>operator<<(std::ostream &strm, const ErrorCluster &err)</code>	ATEsystem_F
<code>ToString() const</code>	ATEsystem_F

Generated by [doxygen](#) 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS > Factory >

ATEsystem_PIRIS::Factory Member List

This is the complete list of members for [ATEsystem_PIRIS::Factory](#), including all inherited members.

CreateDevice (Mode mode=Mode::ETHERNET)	ATEsystem_PIRIS
CreateDevice (const std::string &name, Mode mode)	ATEsystem_PIRIS
Factory (Factory const &)=delete	ATEsystem_PIRIS
GetDeviceInstance (DevID_t id)	ATEsystem_PIRIS
GetDeviceName (DevID_t id)	ATEsystem_PIRIS
getInstance ()	ATEsystem_PIRIS
operator=(Factory const &)=delete	ATEsystem_PIRIS
RemoveDevice (DevID_t id)	ATEsystem_PIRIS
~Factory()	ATEsystem_PIRIS

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS

FocusZoomIris

ATEsystem_PIRIS::FocusZoomIris< T > Member List

This is the complete list of members for **ATEsystem_PIRIS::FocusZoomIris< T >**, including all inherited members.

FocusZoomIris(T focus=NULL, T zoom=NULL, T iris=NULL, int str_pad)
getFocus() const
getFocusZoomIris() const
getIris() const
getZoom() const
operator<<(std::ostream &strm, const FocusZoomIris &vals)
ToString() const

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATESystem_PIRIS

PirisDevice

ATEsystem_PIRIS::PirisDevice Member List

This is the complete list of members for **ATEsystem_PIRIS::PirisDevice**, including all inherited members.

<code>get_camera()</code> const	ATEsystem_F
<code>get_device_eth()</code> const	ATEsystem_F
<code>get_mode()</code> const	ATEsystem_F
<code>get_pylon_dev()</code> const	ATEsystem_F
<code>get_serial()</code> const	ATEsystem_F
<code>get_type()</code> const	ATEsystem_F
<code>operator<<(std::ostream &strm, const PirisDevice &dev)</code>	ATEsystem_F
<code>PirisDevice(Pylon::CDeviceInfo &device)</code>	ATEsystem_F
<code>PirisDevice(Pylon::IPylonDevice *pylon_dev)</code>	ATEsystem_F
<code>PirisDevice(Pylon::CInstantCamera *camera)</code>	ATEsystem_F
<code>PirisDevice(serial::PortInfo &serial)</code>	ATEsystem_F
<code>ToString()</code> const	ATEsystem_F

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATESystem_PIRIS > SmartPointer >

ATEsystem_PIRIS::SmartPointer< T > Member List

This is the complete list of members for **ATEsystem_PIRIS::SmartPointer< T >**, including all inherited members.

isValid() const	ATEsystem_PIRIS::SmartPointer< T >
operator *(void) const	ATEsystem_PIRIS::SmartPointer< T >
operator bool(void) const	ATEsystem_PIRIS::SmartPointer< T >
operator T *(void) const	ATEsystem_PIRIS::SmartPointer< T >
operator()(void) const	ATEsystem_PIRIS::SmartPointer< T >
operator->(void) const	ATEsystem_PIRIS::SmartPointer< T >
operator=(T *pB)	ATEsystem_PIRIS::SmartPointer< T >
operator==(T *pT) const	ATEsystem_PIRIS::SmartPointer< T >
SmartPointer(void) noexcept	ATEsystem_PIRIS::SmartPointer< T >
~SmartPointer(void)	ATEsystem_PIRIS::SmartPointer< T >

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS

StatusEx

ATEsystem_PIRIS::StatusEx Member List

This is the complete list of members for **ATEsystem_PIRIS::StatusEx**, including all inherited members.

ErrorCluster(std::string err_raw="", bool override=false)

getErrNum() const

getErrorCluster() const

getMsg() const

getOk() const

status

StatusEx(Status status=Status::GENERAL_ERROR, ErrorCluster err=E)

ToString() const

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATEsystem_PIRIS > Utils >

ATEsystem_PIRIS::Utils Member List

This is the complete list of members for **ATEsystem_PIRIS::Utils**, including all inherited members.

CopyCStr2STLStr (const char *src)	ATEsys
CopySTLStr2CStr (char *dest, std::string source)	ATEsys
CpyCStr2CStr (char *dest, const char *source, size_t max_len)	ATEsys
PadStr (const int total, const int minus=0, const char ch=' ')	ATEsys

Generated by **doxygen** 1.8.15

ATEsystem.PIRIS-driver



1.0.1

ATEsystem.PIRIS-driver

ATESystem_PIRIS > Version

ATEsystem_PIRIS::Version Member List

This is the complete list of members for [ATEsystem_PIRIS::Version](#), including all inherited members.

getMajor() const	ATEsystem_PIRIS::Version::getMajor
getMinor() const	ATEsystem_PIRIS::Version::getMinor
getRevision() const	ATEsystem_PIRIS::Version::getRevision
getVersion() const	ATEsystem_PIRIS::Version::getVersion
operator<(const Version &ver1, const Version &ver2)	ATEsystem_PIRIS::Version::operator<
operator<<(std::ostream &strm, const Version &ver)	ATEsystem_PIRIS::Version::operator<<
operator<=(const Version &ver1, const Version &ver2)	ATEsystem_PIRIS::Version::operator<=
operator==(const Version &ver1, const Version &ver2)	ATEsystem_PIRIS::Version::operator==
operator>(const Version &ver1, const Version &ver2)	ATEsystem_PIRIS::Version::operator>
operator>=(const Version &ver1, const Version &ver2)	ATEsystem_PIRIS::Version::operator>=
ToString() const	ATEsystem_PIRIS::Version::ToString
Version(uint16_t major=0, uint16_t minor=0, uint16_t revision=0)	ATEsystem_PIRIS::Version::Version