PxVision Ethernet Server
User Guide

www.promax.it

1 PREFACE

This document describes the use of **SERVER ETHERNET** for **PxVision**.
The **Server Ethernet** allows to use **PxVision** by external devices.
For the explanation about PxVision functions, see:

PxVisionBrowser User Guide

1.1 Server Installation

The **Server** is automatically installed by setup PxVision. After executing the setup in to folder where is not needed permissions (ex. C:\PxVision) the system is ready for use.
The Server parameters are set by **PxVisionBrowser**.

1.2 Communication Protocol

The Server uses **TWO COMMUNICATION CHANNELS**, **TCP/IP** port for commands transfer and **UDP** port for Images transfer.
The Server **MUST BE ENABLED** from Configuredon.

Example to server connection in C#

```
Socket _ServerUdp = null;
Socket _ServerTcpIp = null;
IPEndPoint _EndpointUdp = null;
TcpClient _ClientUdp;
/// <summary>
/// Connect the socket
/// <summary>
public bool Connect(string IpAddr)
{
    if (!_VideoPortConnect(IpAddr, 8501))
        return false;
  if (!_CommandPortConnect(IpAddr, 8500))
    return false;
  return true;
}
/// <summary>
/// Udp Connect
/// </summary>
bool _VideoPortConnect(string IpAddr, int VideoPort)
{
  try
  {
    _ServerUdp = new Socket(AddressFamily.InterNetwork, SocketType.Dgram, ProtocolType.Udp);
    _EndpointUdp = new IPEndPoint(IPAddress.Parse(IpAddr), VideoPort);
    _ClientUdp = new TcpClient();
    _ClientUdp.Connect(_EndpointUdp);
    return true;
  }
  catch { return false; }
}

/// <summary>
/// Tcp Ip Connect
/// </summary>
bool _CommandPortConnect(string IpAddr, int CommandPort)
{
  _ServerTcpIp = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
  IPEndPoint _RemoteEndPoint = new IPEndPoint(IPAddress.Parse(IpAddr), CommandPort);
  _ServerTcpIp.ReceiveTimeout = 5000;
  _ServerTcpIp.SendTimeout = 5000;
  try
  {
    _ServerTcpIp.Connect(_RemoteEndPoint);
    return true;
  }
  catch { return false; }
}
```

When the connection is correct the server is ready to use.

1.3 Example of Command and Image Capture

The command will be send by **TCP/IP**.

```
byte[] DataReceived = new byte[20000];

    /// <summary>
    /// Send Command Bar Code Reader
    /// </summary>
    /// <returns></returns>
    bool SendCommand()
    {
       byte[] _Send = new byte[3];
       _Send[0]= 20;    //CMD BAR CODE READER
       _Send[1] = 2;    //EXT CMD GET DATA
       _Send[2] = 2;    //SEND IMAGE
       if (!SendEth(_Send, 3))
          return false;
       int _Ndati = ReadEth();
       if (_Ndati == -1)
          return false;
       if (DataReceived[4] != 20)
          return false;
       GetImage();
    }
    /// <summary>
    /// Send Data
    /// </summary>
    /// <param name="_Val"></param>
    /// <param name="_Len"></param>
    bool SendEth(byte[] _Val, Int32 _Len)
    {
       try
       {
          byte[] _Send = new byte[_Len + 4];
          int _Value = _Len;
          Array.Copy(_Val, 0, _Send, 4, _Len);
          _Send[3] = (byte)_Value;
          _Value <<= 8;
          _Send[2] = (byte)_Value;
          _Value <<= 8;
          _Send[1] = (byte)_Value;
          _Value <<= 8;
          _Send[0] = (byte)_Value;
          _ServerTcpIp.Send(_Send, _Len + 4, SocketFlags.None);
       }
       catch { return false; }
       return true;
    }
```

```
    /// <summary>
    /// Read Data from Ethernet
    /// </summary>
    /// <param name="LenDati"></param>
    /// <returns>array dati</returns>
    internal int ReadEth()
    {
      int _Ndati;
      try
      {
        _Ndati = _ServerTcpIp.Receive(DataReceived, SocketFlags.None);
        DataPointer =0;
        int _Len = GetInt();
        if(_Len!=_Ndati-4)
        {
          int _Diff = (_Ndati - 4) - _Len;
          byte[] BuffRx = new byte[_Diff];
          int _NrDati = 0;
          int _PuntRx = _Ndati;
          while (_PuntRx < _Diff)
          {
            _NrDati = _ServerTcpIp.Receive(BuffRx, _Diff, SocketFlags.None);
            for (int n = 0; n < _NrDati; n++)
              DataReceived[_PuntRx++] = BuffRx[n];
          }
        }
        return _Ndati;
      }

      catch { return -1; }
    }

    /// <summary>
    /// Get Image
    /// </summary>
    /// <returns></returns>
    internal ImageSource GetImage()
    {
      //get image
      NetworkStream _Stream = _ClientUdp.GetStream();
      byte[] _Data = (byte[])_Formatter.Deserialize(_Stream);
      MemoryStream _ImgStream = new MemoryStream(_Data);
      ImageSource _Image = BitmapFrame.Create(_ImgStream, BitmapCreateOptions.None,
BitmapCacheOption.OnLoad);
      return _Image;
    }
```

1.4 TCP/IP Protocol Description

The **TCP/IP** protocol is Bytes mapped, with **COMMANDS** and **DATA**.
For each command sent, follows to reply from the **SERVER**, that contains the data required.
The reply can be also follows da an **IMAGE** sent in **UDP**.
All data values (sent and received) are in **INTEGER BIG ENDIAN,** i.e. **MSB** in the last byte.

Byte 0	Byte 1	Byte 2	Byte 3
LSB	…	…	MSB

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6-…n
LEN_DATA	CMD	CMD_EXT	CMD_DATA

LEN_DATA Integer (4 Bytes) Length data sent from **CMD**
CMD Command
CMD_EXT Command Extension
CMD_DATA Data Command (Length data based on command sent)
Can't be present

Data Received

Byte 0-3	Byte 4	Byte 5-…n
LEN_DATA	CMD	CMD_DATA

LEN_DATA Integer (4 Bytes) Length data sent from **CMD**
CMD **0** COMMAND ERROR – no data - **LEN_DATA=1**
COMMAND OK **CMD**= CMD Sent
CMD_DATA Data Command

Some data sent from and to **PxVision** have the value in **DOUBLE**, these are in any case sent in **INTEGER** with to specific **RATIO**.

VAL_INTEGER=(INTEGER) (VAL_DOUBLE*RATIO)

Example:

VAL_DOUBLE=1.27
RATIO=100
VAL_INTEGER=1.27*100=127

So, when the data are read must be divide for **RATIO**, when are write must multiplied for **RATIO**

Data String
By TCP/IP can be sent/received string of text.
The data contain the ASCII value of text and ends value ZERO (0)
Ex Send/Received "**TEXT**"

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
84 (T)	69 (E)	88 (X)	84 (T)	0

1.5 Command Filtering

Command for **FILTERS** of **PxVision**

Parameters

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
GAUSSIAN	Gaussian - Filter	0	30	1
BLUR	Blur - Filter	0	10	1
MEDIAN BLUR	Median - Filter	0	50	1
LAPLACIAN	Laplacian - Filter	0	16	1
ERODE	Erosion - Filter	0	50	1
DILATE	Dilatation - Filter	0	50	1
GAMMA	Gamma - Filter 2	0	25	**10**
GRAYIMG	Gray/Color - Threshold	0 - Color	1 - Gray	1
THTYPE	Threshold	-1	10	1
THVAL	Tl - Threshold	0	255	1
THMAXVAL	Th - Threshold	0	255	1
ADAPTIVEP1	P1 - Threshold	-40	40	1
CANNY1	C1 - Filter	0	400	1
CANNY2	C2 - Filter	0	400	1
ALPHA	Alpha - Filter 2	0	5	**10**
BETA	Beta - Filter 2	-512	512	1
POS0-POS10	Filter Position	0	10	1

1.5.1 Read Par

Read Parameters **CMD=1 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	**1**	**0**

Data Received 80 bytes (Length included)

Byte 0-3	Byte 4	Byte 5-8	Byte 9-12	Byte 13-16	Byte 17-20	Byte 21-24	Byte 25-28
76	**1/0**	**GAUSSIAN**	**BLUR**	**MEDIAN BLUR**	**LAPLACIAN**	**ERODE**	**DILATE**
Byte 29-32	**Byte 33-36**	**Byte 37-40**	**Byte 41-44**	**Byte 45-48**	**Byte 49-52**	**Byte 53-56**	**Byte 57-60**
GAMMA	**GRAYIMG**	**THTYPE**	**THVAL**	**THMAXVAL**	**ADAPTIVEP1**	**CANNY1**	**CANNY2**
Byte 61-64	**Byte 65-68**	**Byte 69**	**Byte 70**	**Byte 71**	**Byte 72**	**Byte 73**	**Byte 74**
ALPHA	**BETA**	**POS0**	**POS1**	**POS2**	**POS3**	**POS4**	**POS5**
Byte 75	**Byte 76**	**Byte 77**	**Byte 78**	**Byte 79**			
POS6	**POS7**	**POS8**	**POS9**	**POS10**			

1.5.2 Set Par

Write Parameters **CMD=1 CMD_EXT=1**

Data Sent 81 bytes (Length included)

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
77	1	1	GAUSSIAN	BLUR	MEDIAN BLUR	LAPLACIAN	ERODE
Byte 26-29	Byte 30-33	Byte 34-37	Byte 38-41	Byte 42-45	Byte 46-49	Byte 50-53	Byte 54-57
DILATE	GAMMA	GRAYIMG	THTYPE	THVAL	THMAXVAL	ADAPTIVEP1	CANNY1
Byte 58-61	Byte 62-65	Byte 66-69	Byte 70	Byte 71	Byte 72	Byte 73	Byte 74
CANNY2	ALPHA	BETA	POS0	POS1	POS2	POS3	POS4
Byte 75	Byte 76	Byte 77	Byte 78	Byte 79	Byte 80		
POS5	POS6	POS7	POS8	POS9	POS10		

Data Received

Byte 0-3	Byte 4
1	1/0

1.6 Command Square Marker

Command for Marker Square

Parameters

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
X REGION	Camera Region X	0	5000	1
Y REGION	Camera Region Y	0	5000	1
W REGION	Camera Region Width	-1 (Disable)	5000	1
H REGION	Camera Region Height	0	5000	1
LEFT BORD	Border Left	-1 (Disable)	5000	1
RIGHT BORD	Border Right	0	5000	1
TOP BORD	Border Top	0	5000	1
BOT BORD	Border Bottom	0	5000	1
W SQUARE	Width Square	1	3000	1
H SQUARE	Height Square	1	3000	1
ACCURACY	Accuracy	0.001	0.3	**1.000.000**
BLUE BORD	Border Color Blue	0	255	1
GREEN BORD	Border Color Green	0	255	1
RED BORD	Border Color Red	0	255	1
% TOLL	Tolerance	0	100	1
BOUND BOX	Use Bounding Box	0 (No)	1 (Yes)	1
ACQ OK	Acquisition marker Ok	0 (No)	1 (Ok)	1
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
X - CX	X Center Object Detected from camera Center	0	xx	1
Y - CY	Y Center Object Detected from camera Center	0	xx	1
X	X Center Object Detected from camera 0	0	xx	1
Y	Y Center Object Detected from camera 0	0	xx	1
ANGLE	Angle Object Detected Drg	0	xx	**10**
W	Width Object Detected	0	xx	1
H	Heght Object Detected	0	xx	1
X1	X1 Line Bounding Box Object Detected	0	xx	1
Y1	Y1 Line Bounding Box Object Detected	0	xx	1
X2	X2 Line Bounding Box Object Detected	0	xx	1
Y2	Y2 Line Bounding Box Object Detected	0	xx	1
X3	X3 Line Bounding Box Object Detected	0	xx	1
Y3	Y3 Line Bounding Box Object Detected	0	xx	1
X4	X4 Line Bounding Box Object Detected	0	xx	1
Y4	Y4 Line Bounding Box Object Detected	0	xx	1

1.6.1 Read Par

Read Parameters **CMD=2 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	2	0

Data Received 54 bytes (Length included)

Byte 0-3	Byte 4	Byte 5-8	Byte 9-12	Byte 13-16	Byte 17-20	Byte 21-24	Byte 25-28
50	2/0	X REGION	Y REGION	W REGION	H REGION	LEFT BORD	RIGHT BORD
Byte 29-32	Byte 33-36	Byte 37-40	Byte 41-44	Byte 45-48	Byte 49	Byte 50	Byte 51
TOP BORD	BOT BORD	W SQUARE	H SQUARE	ACCURACY	BLUE BORD	GREEN BORD	RED BORD
Byte 52	Byte 53						
% TOLL	BOUND BOX						

1.6.2 Set Par

Write Parameters **CMD=2 CMD_EXT=1**

Data Sent 55 bytes (Length included)

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
51	2	1	X REGION	Y REGION	W REGION	H REGION	LEFT BORD
Byte 26-29	Byte 30-33	Byte 34-37	Byte 38-41	Byte 42-45	Byte 46-49	Byte 50	Byte 51
RIGHT BORD	TOP BORD	BOT BORD	W SQUARE	H SQUARE	ACCURACY	BLUE BORD	GREEN BORD
Byte 52	Byte 53	Byte 54					
RED BORD	% TOLL	BOUND BOX					

Data Received

Byte 0-3	Byte 4
1	2/0

1.6.3 Get Data

Get Data **CMD=2 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	2	2	0(not send) 1(send) 2(send data)

Data Received 70 bytes (Length included)

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
66	2/0	ACQ OK	SCALE	X - CX	Y - CY	X	Y
Byte 26-29	Byte 30-33	Byte 34-37	Byte 38-41	Byte 42-45	Byte 46-49	Byte 50-53	Byte 54-57
ANGLE	W	H	X1	Y1	X2	Y2	X3
Byte 58-61	Byte 62-65	Byte 66-69					
Y3	X4	Y4					

1.7 Command Hole Marker

Command for Marker Hole

Parameters

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
X REGION	Camera Region X	0	5000	1
Y REGION	Camera Region Y	0	5000	1
W REGION	Camera Region Width	-1 (Disable)	5000	1
H REGION	Camera Region Height	0	5000	1
LEFT BORD	Border Left	-1 (Disable)	5000	1
RIGHT BORD	Border Right	0	5000	1
TOP BORD	Border Top	0	5000	1
BOT BORD	Border Bottom	0	5000	1
HOLE DIAM	Diameter Hole	1	3000	1
BLUE BORD	Border Color Blue	0	255	1
GREEN BORD	Border Color Green	0	255	1
RED BORD	Border Color Red	0	255	1
% TOLL	Tolerance	0	100	1
ACQ OK	Acquisition marker Ok	0 (No)	1 (Ok)	1
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
X - CX	X Center Object Detected from camera Center	0	xx	1
Y - CY	Y Center Object Detected from camera Center	0	xx	1
X	X Center Object Detected from camera 0	0	xx	1
Y	Y Center Object Detected from camera 0	0	xx	1
ANGLE	Angle Object Detected Drg	0	xx	**10**
W	Width Object Detected	0	xx	1
H	Heght Object Detected	0	xx	1

1.7.1 Read Par

Read Parameters **CMD=3 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	3	0

Data Received 45 bytes (Length included)

Byte 0-3	Byte 4	Byte 5-8	Byte 9-12	Byte 13-16	Byte 17-20	Byte 21-24	Byte 25-28
41	3/0	X REGION	Y REGION	W REGION	H REGION	LEFT BORD	RIGHT BORD
Byte 29-32	Byte 33-36	Byte 37-40	Byte 41	Byte 42	Byte 43	Byte 44	
TOP BORD	BOT BORD	HOLE DIAM	BLUE BORD	GREEN BORD	RED BORD	% TOLL	

1.7.2 Set Par

Write Parameters **CMD=3 CMD_EXT=1**

Data Sent 46 bytes (Length included)

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
42	3	1	X REGION	Y REGION	W REGION	H REGION	LEFT BORD
Byte 26-29	Byte 30-33	Byte 34-37	Byte 38-41	Byte 42	Byte 43	Byte 44	Byte 45
RIGHT BORD	TOP BORD	BOT BORD	HOLE DIAM	BLUE BORD	GREEN BORD	RED BORD	% TOLL

Data Received

Byte 0-3	Byte 4
1	3/0

1.7.3 Get Data

Get Data **CMD=3 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	3	2	0(not send) 1(send) 2(send data)

Data Received 38 bytes (Length included - Angle non usato)

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
34	3/0	ACQ OK	SCALE	X - CX	Y - CY	X	Y
Byte 26-29	Byte 30-33	Byte 34-37					
ANGLE	W	H					

1.8 Command Find Match

Command for Find Match

Parameters

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
X REGION	Camera Region X	0	5000	1
Y REGION	Camera Region Y	0	5000	1
W REGION	Camera Region Width	-1 (Disable)	5000	1
H REGION	Camera Region Height	0	5000	1
W REG OK	Width Region Ok	0	5000	1
H REG OK	Height Region Ok	0	5000	1
SCOREOK	Score Ok	0	10000	1
%SCOREOK	Delta Score Ok	0	100	1
%REGIONOK	Delta Region Ok	0	100	1
FILENAME	Model Image	String		
ACQ OK	Acquisition marker Ok	0 (No)	1 (Ok)	1
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
X - CX	X Center Object Detected from camera Center	0	xx	1
Y - CY	Y Center Object Detected from camera Center	0	xx	1
X	X Center Object Detected from camera 0	0	xx	1
Y	Y Center Object Detected from camera 0	0	xx	1
ANGLE	Angle Object Detected Drg	0	xx	**10**
W	Width Object Detected	0	xx	1
H	Heght Object Detected	0	xx	1

1.8.1 Read Par

Read Parameters **CMD=4 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
Len=2	4	0

Data Received 35+FileName bytes

Byte 0-3	Byte 4	Byte 5-8	Byte 9-12	Byte 13-16	Byte 17-20	Byte 21-24	Byte 25-28
31+filename	4/0	X REGION	Y REGION	W REGION	H REGION	W REG OK	H REG OK
Byte 29-32	Byte 33	Byte 34	Byte 35				
SCOREOK	%SCOREOK	%REGIONOK	FILENAME 0 End				

1.8.2 Set Par

Write Parameters **CMD=4 CMD_EXT=1**

Data Sent 35 bytes (Length included + Filename)

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
32+filename	4	1	X REGION	Y REGION	W REGION	H REGION	W REG OK
Byte 26-29	Byte 30-33	Byte 34	Byte 35	Byte 36			
H REG OK	SCOREOK	%SCOREOK	%REGIONOK	FILENAME 0 End			

Data Received

Byte 0-3	Byte 4
1	4/0

1.8.3 Get Data

Get Data **CMD=4 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	4	2	0(not send) 1(send) 2(send data)

Data Received 38 bytes (Length included)

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
34	4/0	ACQ OK	SCALE	X - CX	Y - CY	X	Y
Byte 26-29	Byte 30-33	Byte 34-37					
ANGLE	W	H					

1.9 Command Type1 Marker

Command for Marker Type1

Parameters

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
X REGION	Camera Region X	0	5000	1
Y REGION	Camera Region Y	0	5000	1
W REGION	Camera Region Width	-1 (Disable)	5000	1
H REGION	Camera Region Height	0	5000	1
LEFT BORD	Border Left	-1 (Disable)	5000	1
RIGHT BORD	Border Right	0	5000	1
TOP BORD	Border Top	0	5000	1
BOT BORD	Border Bottom	0	5000	1
MARKERDIAM	Marker Diameter	1	3000	1
ACCURACY	Accuracy	0.001	0.3	**1.000.000**
BLUE BORD	Border Color Blue	0	255	1
GREEN BORD	Border Color Green	0	255	1
RED BORD	Border Color Red	0	255	1
% TOLL	Tolerance	0	100	1
%DIAM	Diameter %	10	200	1
NOISE	Pixel Noise	1	100	1
PRESSURE	Pixel Pressure	1	100	1
BLACK	Black Value	0	255	1
WHITE	White Value	0	255	1
ACQ OK	Acquisition marker Ok	0 (No)	1 (Ok)	1
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
X - CX	X Center Object Detected from camera Center	0	xx	1
Y - CY	Y Center Object Detected from camera Center	0	xx	1
X	X Center Object Detected from camera 0	0	xx	1
Y	Y Center Object Detected from camera 0	0	xx	1
ANGLE	Angle Object Detected Drg	0	xx	**10**
W	Width Object Detected	0	xx	1
H	Heght Object Detected	0	xx	1

1.9.1 Read Par

Read Parameters **CMD=5 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	5	0

Data Received 54 bytes (Length included)

Byte 0-3	Byte 4	Byte 5-8	Byte 9-12	Byte 13-16	Byte 17-20	Byte 21-24	Byte 25-28
50	5/0	X REGION	Y REGION	W REGION	H REGION	LEFT BORD	RIGHT BORD
Byte 29-32	Byte 33-36	Byte 37-40	Byte 41-44	Byte 45	Byte 46	Byte 47	Byte 48
TOP BORD	BOT BORD	MARKERDIAM	ACCURACY	BLUE BORD	GREEN BORD	RED BORD	% TOLL
Byte 49	Byte 50	Byte 51	Byte 52	Byte 53			
%DIAM	NOISE	PRESSURE	BLACK	WHITE			

1.9.2 Set Par

Write Parameters **CMD=5 CMD_EXT=1**

Data Sent 55 bytes (Length included)

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
51	5	1	X REGION	Y REGION	W REGION	H REGION	LEFT BORD
Byte 26-29	Byte 30-33	Byte 34-37	Byte 38-41	Byte 42-45	Byte 46	Byte 47	Byte 48
RIGHT BORD	TOP BORD	BOT BORD	MARKERDIAM	ACCURACY	BLUE BORD	GREEN BORD	RED BORD
Byte 49	Byte 50	Byte 51	Byte 52	Byte 53	Byte 54		
% TOLL	%DIAM	NOISE	PRESSURE	BLACK	WHITE		

Data Received

Byte 0-3	Byte 4
1	5/0

1.9.3 Get Data

Get Data **CMD=5 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	5	2	0(not send) 1(send) 2(send data)

Data Received 38 bytes (Length included - Angle non usato)

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
34	5/0	ACQ OK	SCALE	X - CX	Y - CY	X	Y
Byte 26-29	Byte 30-33	Byte 34-37					
ANGLE	W	H					

1.10 Command Circle Marker

Command for Marker Circle

Parameters

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
X REGION	Camera Region X	0	5000	1
Y REGION	Camera Region Y	0	5000	1
W REGION	Camera Region Width	-1 (Disable)	5000	1
H REGION	Camera Region Height	0	5000	1
LEFT BORD	Border Left	-1 (Disable)	5000	1
RIGHT BORD	Border Right	0	5000	1
TOP BORD	Border Top	0	5000	1
BOT BORD	Border Bottom	0	5000	1
MARKERDIAM	Marker Diameter	1	3000	1
ACCURACY	Accuracy	0.001	0.3	**1.000.000**
BLUE BORD	Border Color Blue	0	255	1
GREEN BORD	Border Color Green	0	255	1
RED BORD	Border Color Red	0	255	1
% TOLL	Tolerance	0	100	1
%DIAM	Diameter %	10	200	1
NOISE	Pixel Noise	1	100	1
PRESSURE	Pixel Pressure	1	100	1
BLACK	Black Value	0	255	1
WHITE	White Value	0	255	1
ACQ OK	Acquisition marker Ok	0 (No)	1 (Ok)	1
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
X - CX	X Center Object Detected from camera Center	0	xx	1
Y - CY	Y Center Object Detected from camera Center	0	xx	1
X	X Center Object Detected from camera 0	0	xx	1
Y	Y Center Object Detected from camera 0	0	xx	1
ANGLE	Angle Object Detected Drg	0	xx	**10**
W	Width Object Detected	0	xx	1
H	Heght Object Detected	0	xx	1

1.10.1 Read Par

Read Parameters **CMD=6 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	6	0

Data Received 54 bytes (Length included)

Byte 0-3	Byte 4	Byte 5-8	Byte 9-12	Byte 13-16	Byte 17-20	Byte 21-24	Byte 25-28
50	6/0	X REGION	Y REGION	W REGION	H REGION	LEFT BORD	RIGHT BORD
Byte 29-32	Byte 33-36	Byte 37-40	Byte 41-44	Byte 45	Byte 46	Byte 47	Byte 48
TOP BORD	BOT BORD	MARKERDIAM	ACCURACY	BLUE BORD	GREEN BORD	RED BORD	% TOLL
Byte 49	Byte 50	Byte 51	Byte 52	Byte 53			
%DIAM	NOISE	PRESSURE	BLACK	WHITE			

1.10.2 Set Par

Write Parameters **CMD=6 CMD_EXT=1**

Data Sent 55 bytes (Length included)

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
51	6	1	X REGION	Y REGION	W REGION	H REGION	LEFT BORD
Byte 26-29	Byte 30-33	Byte 34-37	Byte 38-41	Byte 42-45	Byte 46	Byte 47	Byte 48
RIGHT BORD	TOP BORD	BOT BORD	MARKERDIAM	ACCURACY	BLUE BORD	GREEN BORD	RED BORD
Byte 49	Byte 50	Byte 51	Byte 52	Byte 53	Byte 54		
% TOLL	%DIAM	NOISE	PRESSURE	BLACK	WHITE		

Data Received

Byte 0-3	Byte 4
1	6/0

1.10.3 Get Data

Get Data **CMD=6 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	6	2	0(not send) 1(send) 2(send data)

Data Received 38 bytes (Length included - Angle non usato)

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
34	6/0	ACQ OK	SCALE	X - CX	Y - CY	X	Y
Byte 26-29	Byte 30-33	Byte 34-37					
ANGLE	W	H					

1.11 Command Cross Section

Command for Cross Section

Parameters

The Parameters from **StartX** to **Gray** are repeated for the number of Cross Section Configured **CROSS NR**
The data from **ACQ OK** to **YOPEN** are repeated for the number of Cross Section read **CROSS READ**

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
CROSS NR	Number of Cross Section configured	0	20	1
STARTX n	Start X Cross Nr	0	5000	1
STARTY n	Start Y Cross Nr	0	5000	1
ENDX n	End X Cross Nr	0	5000	1
ENDY n	End Y Cross Nr	0	5000	1
INSIDE n	Inside Cross Nr	0 (Outside)	1 (Inside)	1
NOISE n	Pixel Noise Cross Nr	1	100	1
PRESSH n	Pressure High Cross Nr	1	100	1
PRESSL n	Pressure Low Cross Nr	1	100	1
DARK n	Dark Cross Nr	0 (No Dark)	1 (Dark)	1
GRAY n	Gray Value Cross Nr	0	255	1
CROSS READ	Number of Cross Section read	0	20	1
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
ACQ OK n	Acquisition Cross Ok Nr	0 (No)	1 (Ok)	1
MEASURE n	Misura Cross Nr	0	xx	1
XBASE n	X intersection point Base Nr	0	xx	1
YBASE n	Y intersection point Base Nr	0	xx	1
XOPEN n	X intersection point Open Nr	0	xx	1
YOPEN n	Y intersection point Open Nr	0	xx	1

1.11.1 Read Par

Read Parameters **CMD=7 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	7	0

Data Received 6+22*CROSS NR bytes (Length included)
Len=2+22*CROSS NR

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22
Len	7/0	CROSS NR	STARTX n	STARTY n	ENDX n	ENDY n	INSIDE n
Byte 23	Byte 24	Byte 25	Byte 26	Byte 27	Byte 28-31		
NOISE n	PRESSH n	PRESSL n	DARK n	GRAY n	STARTX n+1	…	…

1.11.2 Set Par

Write Parameters **CMD=7 CMD_EXT=1**

Data Sent 7+22*CROSS NR bytes (Length included)
Len=3+22*CROSS NR

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7-10	Byte 11-14	Byte 15-18	Byte 19-21
Len	7	1	CROSS NR	STARTX n	STARTY n	ENDX n	ENDY n
Byte 22	Byte 23	Byte 24	Byte 25	Byte 26	Byte 27	Byte 28-31	
INSIDE n	NOISE n	PRESSH n	PRESSL n	DARK n	GRAY n	STARTX n+1	…

Data Received

Byte 0-3	Byte 4
1	7/0

1.11.3 Get Data

Get Data **CMD=7 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	7	2	0(not send) 1(send) 2(send data)

Data Received 10+25*CROSS READ bytes (Length included)
Len= 6+25*CROSS READ

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10	Byte 11-14	Byte 15-18	Byte 19-22
Len	7/0	CROSS READ	SCALE	ACQ OK n	MEASURE n	XBASE n	Y BASE n
Byte 23-26	Byte 27-30	Byte 31					
XOPEN n	YOPEN n	ACQ OK n+1	…	…	…	…	…

1.12 Command Gap Cross Section

Command for Gap Cross Section

Parameters

The Parameters from **StartX** to **Gray** are repeated for the number of Cross Section Configured **CROSS NR**
The data from **ACQ OK** to **YMAXO** are repeated for the number of Cross Section read **CROSS READ**

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
CROSS NR	Number of Cross Section configured	0	20	1
STARTX n	Start X Cross Nr	0	5000	1
STARTY n	Start Y Cross Nr	0	5000	1
ENDX n	End X Cross Nr	0	5000	1
ENDY n	End Y Cross Nr	0	5000	1
OPENING n	Opening Cross Nr	0	5000	1
GAPMIN n	Min Gap Cross Nr	0	500	1
INSIDE n	Inside Cross Nr	0 (Outside)	1 (Inside)	1
NOISE n	Pixel Noise Cross Nr	1	100	1
PRESSH n	Pressure High Cross Nr	1	100	1
PRESSL n	Pressure Low Cross Nr	1	100	1
DARK n	Dark Cross Nr	0 (No Dark)	1 (Dark)	1
GRAY n	Gray Value Cross Nr	0	255	1
CROSS READ	Number of Cross Section Read	0	20	1
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
ACQ OK n	Acquisition Cross Ok Nr	0 (No)	1 (Ok)	1
MEASMIN n	Misura Min Cross Nr	0	xx	1
MEASMAX n	Misura Max Cross Nr	0	xx	1
XMINB n	X intersection point Base Min Value Nr	0	xx	1
YMINB n	Y intersection point Base Min Value Nr	0	xx	1
XMINO n	X intersection point Open Min Value Nr	0	xx	1
YMINO n	Y intersection point Open Min Value Nr	0	xx	1
XMAXB n	X intersection point Base Max Value Nr	0	xx	1
YMAXB n	Y intersection point Base Max Value Nr	0	xx	1
XMAXO n	X intersection point Open Max Value Nr	0	xx	1
YMAXO n	Y intersection point Open Max Value Nr	0	xx	1

1.12.1 Read Par

Read Parameters **CMD=8 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	8	0

Data Received 6+30*CROSS NR bytes (Length included)
Len=2+30*CROSS NR

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
Len	8/0	CROSS NR	STARTX n	STARTY n	ENDX n	ENDY n	OPENING n
Byte 26-29	Byte 30	Byte 31	Byte 32	Byte 33	Byte 34	Byte 35	Byte 36-39
GAPMIN n	INSIDE n	NOISE n	PRESSH n	PRESSL n	DARK n	GRAY n	STARTX n+1

1.12.2 Set Par

Write Parameters **CMD=8 CMD_EXT=1**

Data Sent 7+30*CROSS NR bytes (Length included)
Len=3+30*CROSS NR

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7-10	Byte 11-14	Byte 15-18	Byte 19-22
Len	8	1	CROSS NR	STARTX n	STARTY n	ENDX n	ENDY n
Byte 23-26	Byte 27-30	Byte 31	Byte 32	Byte 33	Byte 34	Byte 35	Byte 36
OPENING n	GAPMIN n	INSIDE n	NOISE n	PRESSH n	PRESSL n	DARK n	GRAY n
Byte 37-40							
STARTX n+1							

Data Received

Byte 0-3	Byte 4
1	8/0

1.12.3 Get Data

Get Data **CMD=8 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	8	2	0(not send) 1(send) 2(send data)

Data Received 10+45*CROSS READ bytes (Length included)
Len= 6+45*CROSS READ

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10	Byte 11-14	Byte 15-18	Byte 19-22
Len	8/0	CROSS READ	SCALE	ACQ OK n	MEASMIN n	MEASMAX n	XMINB n
Byte 23-26	Byte 27-30	Byte 31-34	Byte 35-38	Byte 39-42	Byte 43-46	Byte 47-50	Byte 51
YMINB n	XMINO n	YMINO n	XMAXB n	YMAXB n	XMAXO n	YMAXO n	ACQ OK n+1

1.13 Command Probe Single Point

Command for Probe Single Point

Parameters

The Parameters from **StartX** to **Gray** are repeated for the number of Probe Configured **PROBE NR**
The data from **ACQ OK** to **Y** are repeated for the number of Probe read **PROBE READ**

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
PROBE NR	Number of Probe configured	0	20	1
STARTX n	Start X Probe Nr	0	5000	1
STARTY n	Start Y Probe Nr	0	5000	1
ENDX n	End X Probe Nr	0	5000	1
ENDY n	End Y Probe Nr	0	5000	1
NOISE n	Pixel Noise Probe Nr	1	100	1
PRESS n	Pressure Probe Nr	1	100	1
DARK n	Dark Probe Nr	0 (No Dark)	1 (Dark)	1
GRAY n	Gray Value Probe Nr	0	255	1
PROBE READ	Number of Probe Read	0	20	1
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
ACQ OK	Acquisition Probe Ok	0 (No)	1 (Ok)	1
X - CX	X Center Object Detected from camera Center	0	xx	1
Y - CY	Y Center Object Detected from camera Center	0	xx	1
X	X Center Object Detected from camera 0	0	xx	1
Y	Y Center Object Detected from camera 0	0	xx	1

1.13.1 Read Par

Read Parameters **CMD=9 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	9	0

Data Received 6+20* PROBE NR bytes (Length included)
Len=2+20* PROBE NR

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22
Len	9/0	PROBE NR	STARTX n	STARTY n	ENDX n	ENDY n	NOISE n
Byte 23	Byte 24	Byte 25	Byte 26-29				
PRESS n	DARK n	GRAY n	STARTX n+1	…	…	…	…

1.13.2 Set Par

Write Parameters **CMD=9 CMD_EXT=1**

Data Sent 7+20* PROBE NR bytes (Length included)
Len=3+20* PROBE NR

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7-10	Byte 11-14	Byte 15-18	Byte 19-22
Len	9	1	PROBE NR	STARTX n	STARTY n	ENDX n	ENDY n
Byte 23	Byte 24	Byte 25	Byte 26	Byte 27-30			
NOISE n	PRESS n	DARK n	GRAY n	STARTX n+1	…	…	…

Data Received

Byte 0-3	Byte 4
1	9/0

1.13.3 Get Data

Get Data **CMD=9 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	9	2	0(not send) 1(send) 2(send data)

Data Received 10+17* PROBE READ bytes (Length included)
Len= 6+17* PROBE READ

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10	Byte 11-14	Byte 15-18	Byte 19-22
Len	9/0	PROBE READ	SCALE	ACQ OK n	X - CX	Y - CY	X
Byte 23-26	Byte 27						
Y	ACQ OK n+1						

1.14 Command Calliper

Command for Calliper

Parameters

The Parameters from **StartX** to **Gray** are repeated for the number of Cross Section Configured **CALL NR**
The data from **ACQ OK** to **YOPEN** are repeated for the number of Cross Section read **CROSS READ**

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
CALL NR	Number of Calliper configured	0	10	1
STARTX n	Start X Calliper Nr	0	5000	1
STARTY n	Start Y Calliper Nr	0	5000	1
ENDX n	End X Calliper Nr	0	5000	1
ENDY n	End Y Calliper Nr	0	5000	1
WIDTH	Base Width Calliper Nr	0	5000	1
INSIDE n	Inside Calliper Nr	0 (Outside)	1 (Inside)	1
NOISE n	Pixel Noise Calliper Nr	1	100	1
PRESSH n	Pressure High Calliper Nr	1	100	1
PRESSL n	Pressure Low Calliper Nr	1	100	1
DARK n	Dark Calliper Nr	0 (No Dark)	1 (Dark)	1
GRAY n	Gray Value Calliper Nr	0	255	1
CALL READ	Number of Calliper read	0	20	1
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
ACQ OK n	Acquisition Calliper Ok Nr	0 (No)	1 (Ok)	1
MEASURE n	Misura Calliper Nr	0	xx	1
XBASE n	X intersection point Base Nr	0	xx	1
YBASE n	Y intersection point Base Nr	0	xx	1
XOPEN n	X intersection point Open Nr	0	xx	1
YOPEN n	Y intersection point Open Nr	0	xx	1

1.14.1 Read Par

Read Parameters **CMD=10 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	10	0

Data Received 6+26* CALL NR bytes (Length included)
Len=2+26* CALL NR

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
Len	10/0	CALL NR	STARTX n	STARTY n	ENDX n	ENDY n	WIDTH
Byte 26	Byte 27	Byte 28	Byte 29	Byte 30	Byte 31	Byte 32-35	
INSIDE n	NOISE n	PRESSH n	PRESSL n	DARK n	GRAY n	STARTX n+1	…

1.14.2 Set Par

Write Parameters **CMD=10 CMD_EXT=1**

Data Sent 7+26* CALL NR bytes (Length included)
Len=3+26* CALL NR

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7-10	Byte 11-14	Byte 15-18	Byte 19-22
Len	10	1	CALL NR	STARTX n	STARTY n	ENDX n	ENDY n
Byte 23-26	Byte 27	Byte 28	Byte 29	Byte 30	Byte 31	Byte 32	Byte 33-36
WIDTH	INSIDE n	NOISE n	PRESSH n	PRESSL n	DARK n	GRAY n	STARTX n+1

Data Received

Byte 0-3	Byte 4
1	10/0

1.14.3 Get Data

Get Data **CMD=10 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	10	2	0(not send) 1(send) 2(send data)

Data Received 10+21* CALL READ bytes (Length included)
Len= 6+21* CALL READ

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10	Byte 11-14	Byte 15-18	Byte 19-22
Len	10/0	CALL READ	SCALE	ACQ OK n	MEASURE n	XBASE n	YBASE n
Byte 23-26	Byte 27-30	Byte 31					
XOPEN n	YOPEN n	ACQ OK n+1					

1.15 Command Probe Line

Command for Probe Line

Parameters

The Parameters from **StartX** to **Gray** are repeated for the number of Probe Configured **PROBE NR**
The data from **ACQ OK** to **Y** are repeated for the number of Probe read **PROBE READ**

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
PROBE NR	Number of Probe configured	0	20	1
STARTX n	Start X Probe Nr	0	5000	1
STARTY n	Start Y Probe Nr	0	5000	1
ENDX n	End X Probe Nr	0	5000	1
ENDY n	End Y Probe Nr	0	5000	1
WIDTH n	Width Probe Nr	0	5000	1
NOISE n	Pixel Noise Probe Nr	1	100	1
PRESS n	Pressure Probe Nr	1	100	1
DARK n	Dark Probe Nr	0 (No Dark)	1 (Dark)	1
GRAY n	Gray Value Probe Nr	0	255	1
PROBE READ	Number of Probe Read	0	20	1
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
ACQ OK	Acquisition Probe Ok	0 (No)	1 (Ok)	1
X – CX	X Center Object Detected from camera Center	0	xx	1
Y – CY	Y Center Object Detected from camera Center	0	xx	1
X	X Center Object Detected from camera 0	0	xx	1
Y	Y Center Object Detected from camera 0	0	xx	1

1.15.1 Read Par

Read Parameters **CMD=11 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	11	0

Data Received 6+24* PROBE NR bytes (Length included)
Len=2+24* PROBE NR

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
Len	11/0	PROBE NR	STARTX n	STARTY n	ENDX n	ENDY n	WIDTH n
Byte 26	Byte 27	Byte 28	Byte 29	Byte 30-33			
NOISE n	PRESS n	DARK n	GRAY n	STARTX n+1	…	…	…

1.15.2 Set Par

Write Parameters **CMD=11 CMD_EXT=1**

Data Sent 7+24* PROBE NR bytes (Length included)
Len=3+24* PROBE NR

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7-10	Byte 11-14	Byte 15-18	Byte 19-22
Len	11	1	PROBE NR	STARTX n	STARTY n	ENDX n	ENDY n
Byte 23-26	Byte 27	Byte 28	Byte 29	Byte 30	Byte 31-34		
WIDTH n	NOISE n	PRESS n	DARK n	GRAY n	STARTX n+1	…	…

Data Received

Byte 0-3	Byte 4
1	11/0

1.15.3 Get Data

Get Data **CMD=11 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	11	2	0(not send) 1(send) 2(send data)

Data Received 10+17* PROBE READ bytes (Length included)
Len= 6+17* PROBE READ

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10	Byte 11-14	Byte 15-18	Byte 19-22
Len	11/0	PROBE READ	SCALE	ACQ OK n	X – CX	Y – CY	X
Byte 23-26	Byte 27						
Y	ACQ OK n+1						

1.16 Command Fit Line

Command for Fit Line

Parameters

The Parameters from **StartX** to **Gray** are repeated for the number of Probe Configured **FIT NR**
The data from **ACQ OK** to **LEN** are repeated for the number of Probe read **FIT READ**

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
FIT NR	Number of FIT configured	0	20	1
STARTX n	Start X FIT Nr	0	5000	1
STARTY n	Start Y FIT Nr	0	5000	1
ENDX n	End X FIT Nr	0	5000	1
ENDY n	End Y FIT Nr	0	5000	1
WIDTH n	Width FIT Nr	0	5000	1
STEP n	Step FIT Nr	0	100	1
NOISE n	Pixel Noise FIT Nr	1	100	1
PRESS n	Pressure FIT Nr	1	100	1
DARK n	Dark FIT Nr	0 (No Dark)	1 (Dark)	1
GRAY n	Gray Value FIT Nr	0	255	1
TYPE n	Type Mode 0 -> Least Square 1 -> First End 2 -> First Middle 3 -> Middle End	0	3	1
FIT READ	Number of FIT Read	0	20	1
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
ACQ OK	Acquisition FIT Ok	0 (No)	1 (Ok)	1
STARTX	X Start Point FIT Detected	0	xx	1
STARTY	Y Start Point FIT Detected	0	xx	1
ENDX	X End Point FIT Detected	0	xx	1
ENDY	Y End Point FIT Detected	0	xx	1
ANGLE	Angle FIT Detected drg	0	xx	**10**
LEN	Len FIT Detected	0	xx	1

1.16.1 Read Par

Read Parameters **CMD=12 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	12	0

Data Received 6+29* FIT NR bytes (Length included)
Len=2+29* FIT NR

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
Len	12/0	FIT NR	STARTX n	STARTY n	ENDX n	ENDY n	WIDTH n
Byte 26-29	Byte 30	Byte 31	Byte 32	Byte 33	Byte 34	Byte 35-38	
STEP n	NOISE n	PRESS n	DARK n	GRAY n	TYPE n	STARTX n+1	…

1.16.2 Set Par

Write Parameters **CMD=12 CMD_EXT=1**

Data Sent 7+29* FIT NR bytes (Length included)
Len=3+29* FIT NR

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7-10	Byte 11-14	Byte 15-18	Byte 19-22
Len	12	1	FIT NR	STARTX n	STARTY n	ENDX n	ENDY n
Byte 23-26	Byte 27-30	Byte 31	Byte 32	Byte 33	Byte 34	Byte 35	Byte 36-39
WIDTH n	STEP n	NOISE n	PRESS n	DARK n	GRAY n	TYPE n	STARTX n+1

Data Received

Byte 0-3	Byte 4
1	12/0

1.16.3 Get Data

Get Data **CMD=12 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	12	2	0(not send) 1(send) 2(send data)

Data Received 10+25* FIT READ bytes (Length included)
Len= 6+25* FIT READ

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10	Byte 11-14	Byte 15-18	Byte 19-22
Len	12/0	FIT READ	SCALE	ACQ OK n	STARTX n	STARTY	ENDX
Byte 23-26	Byte 27-30	Byte 31-34	Byte 35				
ENDY	ANGLE n	LEN n	ACQ OK n+1	…	…	…	…

1.17 Command Fit Circle

Command for Fit Circle

Parameters

The Parameters from **CX** to **Gray** are repeated for the number of Probe Configured **FIT NR**
The data from **ACQ OK** to **CY** are repeated for the number of Probe read **FIT READ**

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
FIT NR	Number of FIT configured	0	20	1
CX n	Center X FIT Nr	0	5000	1
CY n	Center Y FIT Nr	0	5000	1
EXTRAD n	External Radius FIT Nr	0	5000	1
INTRAD n	Internal Radius FIT Nr	0	5000	1
SANG n	Start Angle FIT Nr	0	5000	1
EANG n	End Angle FIT Nr	0	5000	1
STEP n	Angle Step FIT Nr	0.1	20	**100**
ACQDIR n	Acq Direction FIT Nr	0 (Ext to Int)	1 (Int to Ext)	1
TYPE n	Type Mode	0 (Least Square)	1 (3 Points)	1
NOISE n	Pixel Noise FIT Nr	1	100	1
PRESS n	Pressure FIT Nr	1	100	1
DARK n	Dark FIT Nr	0 (No Dark)	1 (Dark)	1
GRAY n	Gray Value FIT Nr	0	255	1
FIT READ	Number of FIT Read	0	20	1
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
ACQ OK	Acquisition FIT Ok	0 (No)	1 (Ok)	1
STARTX	X Start Point FIT Detected	0	xx	1
STARTY	Y Start Point FIT Detected	0	xx	1
ENDX	X End Point FIT Detected	0	xx	1
ENDY	Y End Point FIT Detected	0	xx	1
RADIUS	Radius FIT Detected	0	xx	**100**
CX	X Center FIT Detected	0	xx	1
CY	Y Center FIT Detected	0	xx	1

1.17.1 Read Par

Read Parameters **CMD=13 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	13	0

Data Received 6+34* FIT NR bytes (Length included)
Len=2+34* FIT NR

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
Len	13/0	FIT NR	CX n	CY n	EXTRAD n	INTRAD n	SANG n
Byte 26-29	Byte 30-33	Byte 34	Byte 35	Byte 36	Byte 37	Byte 38	Byte 39
EANG n	STEP n	ACQDIR n	TYPE n	NOISE n	PRESS n	DARK n	GRAY n
Byte 40-43							
CX n+1							

1.17.2 Set Par

Write Parameters **CMD=13 CMD_EXT=1**

Data Sent 7+34* FIT NR bytes (Length included)
Len=3+34* FIT NR

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7-10	Byte 11-14	Byte 15-18	Byte 19-22
Len	13	1	FIT NR	CX n	CY n	EXTRAD n	INTRAD n
Byte 23-26	Byte 27-30	Byte 31-34	Byte 35	Byte 36	Byte 37	Byte 38	Byte 39
SANG n	EANG n	STEP n	ACQDIR n	TYPE n	NOISE n	PRESS n	DARK n
Byte 40	Byte 41-44						
GRAY n	CX n+1						

Data Received

Byte 0-3	Byte 4
1	13/0

1.17.3 Get Data

Get Data **CMD=13 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	13	2	0(not send) 1(send) 2(send data)

Data Received 10+29* FIT READ bytes (Length included)
Len= 6+29* FIT READ

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10	Byte 11-14	Byte 15-18	Byte 19-22
Len	13/0	FIT READ	SCALE	ACQ OK n	STARTX n	STARTY n	ENDX n
Byte 23-26	Byte 27-30	Byte 31-34	Byte 35-38	Byte 39			
ENDY n	RADIUS n	CX n	CY n	ACQ OK n+1	…	…	…

1.18 Command Gray Detector

Command for Gray Detector

Parameters

The Parameters from **REG[0]** to **RGBCH** are repeated for the number of Detector Configured **GRAY NR**
The Data **ACQ OK** are repeated for the number of Detector read **GRAY READ**

Type Region=Rectangle

REG[0]=X
REG[1]=Y
REG[2]=W
REG[3]=H

Type Region=Circle

REG[0]=Cx
REG[1]=Cy
REG[2]=R
REG[3]=*** (not used)

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
GRAY NR	Number of Gray configured	0	20	1
REG[0] n	Par Region GRAY Nr	0	5000	1
REG[1] n	Par Region GRAY Nr	0	5000	1
REG[2] n	Par Region GRAY Nr	0	5000	1
REG[3] n	Par Region GRAY Nr	0	5000	1
REGTYPE n	Region Type GRAY Nr	0 (Rectangle)	1 (Circle)	1
GRAYMIN n	Gray Min GRAY Nr	0	255	1
GRAYMAX n	Gray Max GRAY Nr	0	255	1
THMIN n	TH Min GRAY Nr	0	100	1
THMAX n	TH Max GRAY Nr	0	100	1
RGBCH n	Rgb Channel 0 -> Gray 1 -> Red 2 -> Green 3 -> Blue	0	3	1
GRAY READ	Number of GRAY Read	0	20	1
ACQ OK	Acquisition GRAY Ok	0 (No)	1 (Ok)	1

1.18.1 Read Par

Read Parameters **CMD=14 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	14	0

Data Received 6+22* GRAY NR bytes (Length included)
Len=2+22* GRAY NR

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22
Len	14/0	GRAY NR	REG[0] n	REG[1] n	REG[2] n	REG[3] n	REGTYPE n
Byte 23	Byte 24	Byte 25	Byte 26	Byte 27	Byte 28-31		
GRAYMIN n	GRAYMAX n	THMIN n	THMAX n	RGBCH n	REG[0] n+1	…	…

1.18.2 Set Par

Write Parameters **CMD=14 CMD_EXT=1**

Data Sent 7+22* GRAY NR bytes (Length included)
Len=3+22* GRAY NR

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7-10	Byte 11-14	Byte 15-18	Byte 19-22
Len	14	1	GRAY NR	REG[0] n	REG[1] n	REG[2] n	REG[3] n
Byte 23	Byte 24	Byte 25	Byte 26	Byte 27	Byte 28	Byte 29-32	
REGTYPE n	GRAYMIN n	GRAYMAX n	THMIN n	THMAX n	RGBCH n	REG[0] n+1	

Data Received

Byte 0-3	Byte 4
1	14/0

1.18.3 Get Data

Get Data **CMD=14 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	14	2	0(not send) 1(send) 2(send data)

Data Received 6+GRAY READ bytes (Length included)
Len= 2+ GRAY READ

Data Received .. bytes Len=2+Gray Nr.

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7			
Len	14/0	GRAY READ	ACQ OK n	ACQ OK n+1	…	…	…

1.19 Command Contrast Detector

Command for Contrast Detector

Parameters

The Parameters from **REG[0]** to **RGBCH** are repeated for the number of Detector Configured **CONTR NR**
The Data **ACQ OK** are repeated for the number of Detector read **CONTR READ**

Type Region=Rectangle

REG[0]=X
REG[1]=Y
REG[2]=W
REG[3]=H

Type Region=Circle

REG[0]=Cx
REG[1]=Cy
REG[2]=R
REG[3]=*** (not used)

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
CONTR NR	Number of Contrast configured	0	20	1
REG[0] n	Par Region CONTR Nr	0	5000	1
REG[1] n	Par Region CONTR Nr	0	5000	1
REG[2] n	Par Region CONTR Nr	0	5000	1
REG[3] n	Par Region CONTR Nr	0	5000	1
REGTYPE n	Region Type CONTR Nr	0 (Rectangle)	1 (Circle)	1
THMIN n	TH Min CONTR Nr	0	100	1
THMAX n	TH Max CONTR Nr	0	100	1
RGBCH n	Rgb Channel 0 -> Gray 1 -> Red 2 -> Green 3 -> Blue	0	3	1
CONTR READ	Number of CONTR Read	0	20	1
ACQ OK	Acquisition CONTR Ok	0 (No)	1 (Ok)	1

1.19.1 Read Par

Read Parameters **CMD=15 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	15	0

Data Received 6+20* CONTR NR bytes (Length included)
Len=2+20* CONTR NR

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22
Len	15/0	CONTR NR	REG[0] n	REG[1] n	REG[2] n	REG[3] n	REGTYPE n
Byte 23	Byte 24	Byte 25	Byte 26-29				
THMIN n	THMAX n	RGBCH n	REG[0] n+1				

1.19.2 Set Par

Write Parameters **CMD=15 CMD_EXT=1**

Data Sent 7+20* CONTR NR bytes (Length included)
Len=3+20* CONTR NR

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7-10	Byte 11-14	Byte 15-18	Byte 19-22
Len	15	1	GRAY NR	REG[0] n	REG[1] n	REG[2] n	REG[3] n
Byte 23	Byte 24	Byte 25	Byte 26	Byte 27-30			
REGTYPE n	THMIN n	THMAX n	RGBCH n	REG[0] n+1			

Data Received

Byte 0-3	Byte 4
1	15/0

1.19.3 Get Data

Get Data **CMD=15 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	15	2	0(not send) 1(send) 2(send data)

Data Received 6+ CONTR READ bytes (Length included)
Len= 2+ CONTR READ

Data Received .. bytes Len=2+Gray Nr.

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7			
Len	15/0	CONTR READ	ACQ OK n	ACQ OK n+1	…	…	…

1.20 Command Brightness Detector

Command for Brightness Detector

Parameters

The Parameters from **REG[0]** to **RGBCH** are repeated for the number of Detector Configured **CONTR NR**
The Data **ACQ OK** are repeated for the number of Detector read **CONTR READ**

Type Region=Rectangle

REG[0]=X
REG[1]=Y
REG[2]=W
REG[3]=H

Type Region=Circle

REG[0]=Cx
REG[1]=Cy
REG[2]=R
REG[3]=*** (not used)

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
BRH NR	Number of Brightness configured	0	20	1
REG[0] n	Par Region BRH Nr	0	5000	1
REG[1] n	Par Region BRH Nr	0	5000	1
REG[2] n	Par Region BRH Nr	0	5000	1
REG[3] n	Par Region BRH Nr	0	5000	1
REGTYPE n	Region Type BRH Nr	0 (Rectangle)	1 (Circle)	1
THMIN n	TH Min BRH Nr	0	100	1
THMAX n	TH Max BRH Nr	0	100	1
RGBCH n	Rgb Channel 0 -> Gray 1 -> Red 2 -> Green 3 -> Blue	0	3	1
BRH READ	Number of BRH Read	0	20	1
ACQ OK	Acquisition BRH Ok	0 (No)	1 (Ok)	1

1.20.1 Read Par

Read Parameters **CMD=16 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	16	0

Data Received 6+20* BRH NR bytes (Length included)
Len=2+20* BRH NR

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22
Len	16/0	BRH NR	REG[0] n	REG[1] n	REG[2] n	REG[3] n	REGTYPE n
Byte 23	Byte 24	Byte 25	Byte 26-29				
THMIN n	THMAX n	RGBCH n	REG[0] n+1				

1.20.2 Set Par

Write Parameters **CMD=16 CMD_EXT=1**

Data Sent 7+20* BRH NR bytes (Length included)
Len=3+20* BRH NR

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7-10	Byte 11-14	Byte 15-18	Byte 19-22
Len	16	1	GRAY NR	REG[0] n	REG[1] n	REG[2] n	REG[3] n
Byte 23	Byte 24	Byte 25	Byte 26	Byte 27-30			
REGTYPE n	THMIN n	THMAX n	RGBCH n	REG[0] n+1			

Data Received

Byte 0-3	Byte 4
1	16/0

1.20.3 Get Data

Get Data **CMD=16 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	16	2	0(not send) 1(send) 2(send data)

Data Received 6+ BRH READ bytes (Length included)
Len= 2+ BRH READ

Data Received .. bytes Len=2+Gray Nr.

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7			
Len	16/0	BRH READ	ACQ OK n	ACQ OK n+1	...	...	...

1.21 Command Objects Color Detector

Command for Objects Color Detector

Parameters

The Parameters from **XREGION** to **TYPEAREA** are repeated for the number of Detector Configured **OBJ NR**
The data from **OBJ SET** ad **ANGLE** are repeated for the number of Detector Configured **OBJ NR**
The data from **XB** to **ANGLE** are repeated for the number of objects detected **OBJ DET**

TCP/IP	PxVisionBrowser	Val Min	Val Max	Rapp
OBJ NR	Number of Object configured	0	10	1
XREGION n	X Region OBJ Nr	0	5000	1
YREGION n	Y Region OBJ Nr	0	5000	1
WREGION n	W Region OBJ Nr	0	5000	1
HREGION n	H Region OBJ Nr	0	5000	1
LEFT BORD	Border Left	-1 (Disable)	5000	1
RIGHT BORD	Border Right	0	5000	1
TOP BORD	Border Top	0	5000	1
BOT BORD	Border Bottom	0	5000	1
MINAREA n	Min Area OBJ Nr	0	300000	1
MAXAREA n	Max Area OBJ Nr	0	5000000	1
WRECT n	Width Rectangle OBJ Nr	0	5000	1
HRECT n	Height Rectangle OBJ Nr	0	5000	1
DIAM n	Diameter Circle OBJ Nr	0	2500	1
HMIN n	H Min OBJ Nr	0	360	1
HMAX n	H Max OBJ Nr	0	360	1
SMIN n	S Min OBJ Nr	0	255	1
SMAX n	S Max OBJ Nr	0	255	1
VMIN n	V Min OBJ Nr	0	255	1
VMAX n	V Max OBJ Nr	0	255	1
BLUE BORD	Border Color Blue	0	255	1
GREEN BORD	Border Color Green	0	255	1
RED BORD	Border Color Red	0	255	1
RECTW% n	Tolerance Width Rectangle OBJ Nr	0	100	1
RECTH% n	Tolerance Height Rectangle OBJ Nr	0	100	1
DIAM% n	Tolerance Circle OBJ Nr	0	100	1
TYPEAREA n	Type Area OBJ Nr 0 -> Free 1 -> Rectangle 2 -> Circle	0	2	1
OBJ SET	Number of OBJ Setted	0	10	1
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
OBJ DET	Number of Objects Detected	0	xx	1
XB	X Bounding Box for Obj	0	xx	1
YB	Y Bounding Box for Obj	0	xx	1
WB	W Bounding Box for Obj	0	xx	1
HB	H Bounding Box for Obj	0	xx	1
X1R,Y1R, X2R,Y2R X3R,Y3R X4R,Y4R	Edge for Rectangle OBJ	0	xx	1
X - CX	X Center Object Detected from camera Center	0	xx	1
Y - CY	Y Center Object Detected from camera Center	0	xx	1
X	X Center Object Detected from camera 0	0	xx	1
Y	Y Center Object Detected from camera	0	xx	1

	0			
AREAGET	Area Get for Object Detected			
ANGLE	Angle Object Detected Double.MinValue angle not used	0	xx	**100**
XB n+1	Next Objecte Detected if OBJ DET > 1	…	…	**…**
YB n+1	…			**…**
…	…	…	…	**…**

1.21.1 Read Par

Read Parameters **CMD=17 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	17	0

Data Received 6+71* OBJ NR bytes (Length included)
Len=2+71* OBJ NR

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
Len	17/0	OBJ NR	XREGION n	YREGION n	WREGION n	HREGION n	LEFT B n
Byte 26-29	Byte 30-33	Byte 34-37	Byte 38-41	Byte 42-45	Byte 46-49	Byte 50-53	Byte 54-57
RIGHTB n	TOPB n	BTMB n	MINAREA n	MAXAREA n	WRECT n	HRECT n	DIAM n
Byte 58-61	Byte 62-65	Byte 66	Byte 67	Byte 68	Byte 69	Byte 70	Byte 71
HMIN n	HMAX n	SMIN n	SMAX n	VMIN n	VMAX n	B BORD n	G BORD n
Byte 72	Byte 73	Byte 74	Byte 75	Byte 76	Byte 77		
R BORD n	RECTW% n	RECTH %n	DIAM %n	TYPEAREA n	XREGION n+1		

1.21.2 Set Par

Write Parameters **CMD=17 CMD_EXT=1**

Data Sent 7+71* OBJ NR bytes (Length included)
Len=3+71* OBJ NR

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7-10	Byte 11-14	Byte 15-18	Byte 19-22
Len	17	1	OBJ NR	XREGION n	YREGION n	WREGION n	HREGION n
Byte 23-26	Byte 27-30	Byte 31-34	Byte 35-38	Byte 39-42	Byte 43-46	Byte 47-50	Byte 51-54
LEFT B n	RIGHTB n	TOPB n	BTMB n	MINAREA n	MAXAREA n	WRECT n	HRECT n
Byte 55-58	Byte 59-62	Byte 63-65	Byte 66	Byte 67	Byte 68	Byte 69	Byte 70
DIAM n	HMIN n	HMAX n	SMIN n	SMAX n	VMIN n	VMAX n	B BORD n
Byte 71	Byte 72	Byte 73	Byte 74	Byte 75	Byte 76	Byte 77-80	
G BORD n	R BORD n	RECTW% n	RECTH %n	DIAM %n	TYPEAREA n	XREGION n+1	

Data Received

Byte 0-3	Byte 4
1	17/0

1.21.3 Get Data

Get Data **CMD=17 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	17	2	0(not send) 1(send) 2(send data)

Data Received 14+ OBJ SET*(Obj Det*62) (Length included)
Len= 10+ OBJ SET*(Obj Det*62)

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
Len	17/0	OBJ SET	SCALE	OBJ DET n	XB n	YB n	WB n
Byte 26-29	Byte 30-33	Byte 34-37	Byte 38-41	Byte 42-45	Byte 48-51	Byte 52-55	Byte 56-59
HB n	X1R n	Y1R n	X2R n	Y2R n	X3R n	Y3R n	X4R n
Byte 60-63	Byte 64-67	Byte 68-71	Byte 72-75	Byte 76-79	Byte 80-83	Byte 84-87	Byte 88-91
Y4R n	X – CX n	Y – CY n	X n	Y n	AREAGET n	ANGLE n	Next

1.22 Command Objects Area Detector

Command for Objects Area Detector

Parameters

The Parameters from **XREGION** to **TYPEAREA** are repeated for the number of Detector Configured **OBJ NR**
The data from **OBJ SET** ad **ANGLE** are repeated for the number of Detector Configured **OBJ NR**
The data from **XB** to **ANGLE** are repeated for the number of objects detected **OBJ DET**

TCP/IP	PxVisionBrowser	Val Min	Val Max	Rapp
OBJ NR	Number of Object configured	0	10	1
XREGION n	X Region OBJ Nr	0	5000	1
YREGION n	Y Region OBJ Nr	0	5000	1
WREGION n	W Region OBJ Nr	0	5000	1
HREGION n	H Region OBJ Nr	0	5000	1
LEFT BORD	Border Left	-1 Disable	5000	1
RIGHT BORD	Border Right	0	5000	1
TOP BORD	Border Top	0	5000	1
BOT BORD	Border Bottom	0	5000	1
MINAREA n	Min Area OBJ Nr	0	300000	1
MAXAREA n	Max Area OBJ Nr	0	5000000	1
WRECT n	Width Rectangle OBJ Nr	0	5000	1
HRECT n	Height Rectangle OBJ Nr	0	5000	1
DIAM n	Diameter Circle OBJ Nr	0	2500	1
BLUE BORD	Border Color Blue	0	255	1
GREEN BORD	Border Color Green	0	255	1
RED BORD	Border Color Red	0	255	1
RECTW% n	Tolerance Width Rectangle OBJ Nr	0	100	1
RECTH% n	Tolerance Heght Rectangle OBJ Nr	0	100	1
DIAM% n	Tolerance Circle OBJ Nr	0	100	1
TYPEAREA n	Type Area OBJ Nr 0 -> Free 1 -> Rectangle 2 -> Circle	0	2	1
OBJ SET	Number of OBJ Setted	0	10	1
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
OBJ DET	Number of Objects Detected	0	xx	1
XB	X Bounding Box for Obj	0	xx	1
YB	Y Bounding Box for Obj	0	xx	1
WB	W Bounding Box for Obj	0	xx	1
HB	H Bounding Box for Obj	0	xx	1
X1R,Y1R, X2R,Y2R X3R,Y3R X4R,Y4R	Edge for Rectangle OBJ	0	xx	1
X - CX	X Center Object Detected from camera Center	0	xx	1
Y - CY	Y Center Object Detected from camera Center	0	xx	1
X	X Center Object Detected from camera 0	0	xx	1
Y	Y Center Object Detected from camera 0	0	xx	1
AREAGET	Area Get for Object Detected			
ANGLE	Angle Object Detected Double.MinValue angle not used	0	xx	**100**
XB n+1	Next Objecte Detected if OBJ DET > 1	…	…	**…**
YB n+1	…			**…**

1.22.1 Read Par

Read Parameters **CMD=18 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	18	0

Data Received 6+59* OBJ NR bytes (Length included)
Len=2+59* OBJ NR

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
Len	18/0	OBJ NR	XREGION n	YREGION n	WREGION n	HREGION n	LEFTB n
Byte 26-29	Byte 30-33	Byte 34-37	Byte 38-41	Byte 42-45	Byte 46-49	Byte 50-53	Byte 54-57
RIGHTB n	TOPB n	BTMB n	MINAREA n	MAXAREA n	WRECT n	HRECT n	DIAM n
Byte 58	Byte 59	Byte 60	Byte 61	Byte 62	Byte 63	Byte 64	Byte 65
B BORD n	G BORD n	R BORD n	RECTW% n	RECTH %n	DIAM %n	TYPEAREA n	

1.22.2 Set Par

Write Parameters **CMD=18 CMD_EXT=1**

Data Sent 7+59* OBJ NR bytes (Length included)
Len=3+59* OBJ NR

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7-10	Byte 11-14	Byte 15-18	Byte 19-22
Len	18	1	OBJ NR	XREGION n	YREGION n	WREGION n	HREGION n
Byte 23-26	Byte 27-30	Byte 31-34	Byte 35-38	Byte 39-42	Byte 43-46	Byte 47-50	Byte 51-54
LEFTB n	RIGHTB n	TOPB n	BTMB n	MINAREA n	MAXAREA n	WRECT n	HRECT n
Byte 55-58	Byte 59	Byte 60	Byte 61	Byte 62	Byte 63	Byte 64	Byte 65
DIAM n	B BORD n	G BORD n	R BORD n	RECTW% n	RECTH% n	DIAM %n	TYPEAREA n
Byte 66-69							
XREGION n+1							

Data Received

Byte 0-3	Byte 4
1	18/0

1.22.3 Get Data

Get Data **CMD=18 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	18	2	0(not send) 1(send) 2(send data)

Data Received 14+ OBJ SET*(Obj Det*62) (Length included)
Len= 10+ OBJ SET*(Obj Det*62)

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
Len	18/0	OBJ SET	SCALE	OBJ DET n	XB n	YB n	WB n
Byte 26-29	Byte 30-33	Byte 34-37	Byte 38-41	Byte 42-45	Byte 48-51	Byte 52-55	Byte 56-59
HB n	X1R n	Y1R n	X2R n	Y2R n	X3R n	Y3R n	X4R n
Byte 60-63	Byte 64-67	Byte 68-71	Byte 72-75	Byte 76-79	Byte 80-83	Byte 84-87	Byte 88-91
Y4R n	X – CX n	Y – CY n	X n	Y n	AREAGET n	ANGLE n	Next

1.23 Command Image Mask

Command for Mask

Parameters

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
XREGION	X Region	0	5000	1
YREGION	Y Region	0	5000	1
WREGION	W Region	0	5000	1
HREGION	H Region	0	5000	1
ENABLE	Enable Mask	0 (Disable)	1 (Enable)	1
MODEFRM	Mode Frame 0 -> Disable 1 -> Continuous 2 -> Single	0	2	1
MODEMSK	Mask Mode 0 -> AbsDif 1 -> Add 2 -> BitWiseAnd 3 -> BitWiseOr 4 -> BitWiseXor 5 -> BitWiseNot 6 -> Max 7 -> Min 8 -> Subract 9 -> CompareEqual 10 -> CompareGreaterEqual 11 -> CompareGreaterThan 12 -> CompareLessEqual 13 -> CompareLessThan 14 -> CompareNotEqual	0	14	1
MASKNAME	Mask File Name	string		

1.23.1 Read Par

Read Parameters **CMD=19 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	19	0

Data Received 24 + MaskName Length bytes (Length included)
Len=20 + MaskName Length bytes

Byte 0-3	Byte 4	Byte 5-8	Byte 9-12	Byte 13-16	Byte 17-20	Byte 21	Byte 22
Len	19/0	XREGION	YREGION	WREGION	HREGION	ENABLE	MODEFRM
Byte 23	Byte 24 …						
MODEMSK	MASKNAME (0 end of string)						

1.23.2 Set Par

Write Parameters **CMD=19 CMD_EXT=1**

Data Sent 25 + MaskName Length bytes (Length included)
Len=21 + MaskName Length bytes

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22
Len	19	1	XREGION	YREGION	WREGION	HREGION	ENABLE
Byte 23	Byte 24	Byte 25 …					
MODEFRM	MODEMSK	MASKNAME (0 end of string)					

Data Received

Byte 0-3	Byte 4
1	19/0

1.24 Command BarCode Reader

Command for BarCode

Parameters

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
X REGION	Camera Region X	0	5000	1
Y REGION	Camera Region Y	0	5000	1
W REGION	Camera Region Width	-1 (Disable)	5000	1
H REGION	Camera Region Height	0	5000	1
LEFT BORD	Border Left	-1 (Disable)	5000	1
RIGHT BORD	Border Right	0	5000	1
TOP BORD	Border Top	0	5000	1
BOT BORD	Border Bottom	0	5000	1
BLUE BORD	Border Color Blue	0	255	1
GREEN BORD	Border Color Green	0	255	1
RED BORD	Border Color Red	0	255	1
AUTOROTATE	AutoRotate	0 (Disabled)	1 (Enabled)	1
ACQ OK	Acquisition BarcOde Ok	0 (No)	1 (Ok)	1
FORMAT	BarCode Format	0	xx	1
CODE	Code Read	string		

1.24.1 Read Par

Read Parameters **CMD=20 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	20	0

Data Received 41 bytes (Length included)

Byte 0-3	Byte 4	Byte 5-8	Byte 9-12	Byte 13-16	Byte 17-20	Byte 21-24	Byte 25-28
37	20/0	X REGION	Y REGION	W REGION	H REGION	LEFT BORD	RIGHT BORD
Byte 29-32	Byte 33-36	Byte 37	Byte 38	Byte 39	Byte 40		
TOP BORD	BOT BORD	BLUE BORD	GREEN BORD	RED BORD	AUTOROTATE		

1.24.2 Set Par

Write Parameters **CMD=20 CMD_EXT=1**

Data Sent 55 bytes (Length included)

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
38	20	1	X REGION	Y REGION	W REGION	H REGION	LEFT BORD
Byte 26-29	Byte 30-33	Byte 34-37	Byte 38	Byte 39	Byte 40	Byte 41	
RIGHT BORD	TOP BORD	BOT BORD	BLUE BORD	GREEN BORD	RED BORD	AUTOROTATE	

Data Received

Byte 0-3	Byte 4
1	20/0

1.24.3 Get Data

Get Data **CMD=20 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	20	2	0(not send) 1(send) 2(send data)

Data Received 7+Code.Length bytes (Length included)
Len=3+Code.Length

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7
Len	20/0	ACQ OK	FORMAT	Code (0 end of string)

1.25 Command OCR Reader

Command for OCR Reader

Parameters

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
X REGION	Camera Region X	0	5000	1
Y REGION	Camera Region Y	0	5000	1
W REGION	Camera Region Width	-1 (Disable)	5000	1
H REGION	Camera Region Height	0	5000	1
LEFT BORD	Border Left	-1 (Disable)	5000	1
RIGHT BORD	Border Right	0	5000	1
TOP BORD	Border Top	0	5000	1
BOT BORD	Border Bottom	0	5000	1
BLUE BORD	Border Color Blue	0	255	1
GREEN BORD	Border Color Green	0	255	1
RED BORD	Border Color Red	0	255	1
ENGINE	Engine 0 -> TesseractOnly 1 -> LSTMOnly 2 -> TESSERACTLSTMCombined 3 -> Default	0	3	1
SCORE OK	Score Ok %	0	100	1
LANG	Language ita eng deu deu_frak fra	string		
ACQ OK	Acquisition BarcOde Ok	0 (No)	1 (Ok)	1
SCORE	Score Read %	0	100	1
CODE	Code Read	string		

1.25.1 Read Par

Read Parameters **CMD=21 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	21	0

Data Received 42+Lang.length bytes (Length included)
Len=38 + Lang.length

Byte 0-3	Byte 4	Byte 5-8	Byte 9-12	Byte 13-16	Byte 17-20	Byte 21-24	Byte 25-28
Len	21/0	X REGION	Y REGION	W REGION	H REGION	LEFT BORD	RIGHT BORD
Byte 29-32	Byte 33-36	Byte 37	Byte 38	Byte 39	Byte 40	Byte 41	Byte 42
TOP BORD	BOT BORD	BLUE BORD	GREEN BORD	RED BORD	ENGINE	SCORE OK	LANG

1.25.2 Set Par

Write Parameters **CMD=21 CMD_EXT=1**

Data Sent 43+Lang.length bytes (Length included)
Len=39 + Lang.length

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
Len	21	1	X REGION	Y REGION	W REGION	H REGION	LEFT BORD
Byte 26-29	Byte 30-33	Byte 34-37	Byte 38	Byte 39	Byte 40	Byte 41	Byte 42
RIGHT BORD	TOP BORD	BOT BORD	BLUE BORD	GREEN BORD	RED BORD	ENGINE	SCORE OK
Byte 42							
LANG (0 end of string)							

Data Received

Byte 0-3	Byte 4
1	21/0

1.25.3 Get Data

Get Data **CMD=21 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	21	2	0(not send) 1(send) 2(send data)

Data Received 7+Code.Length bytes (Length included)
Len=3+Code.Length

Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7
Len	21/0	ACQ OK	SCORE	Code (0 end of string)

1.26 Command Camera Setting

Command for Camera Parameters

Parameters

The Parameters with ValMin ValMax = DP are device dipendent

Referred to Val Min and Max used from PxVisionBrowser

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
SCALE	Scale Factor Pixel/MM	0	10	**100.000.000**
RANGLE	Rotation Angle drg	0	360	**100**
RCX	Rotation Center X	0	5000	1
RCY	Rotation Center Y	0	5000	1
BRIGHT	Brightness	**DP**	**DP**	1
SAT	Saturation	**DP**	**DP**	1
CONT	Contrast	**DP**	**DP**	1
SHARP	Sharpness	**DP**	**DP**	1
GAMMA	Gamma	**DP**	**DP**	1
BACK	BackLight	**DP**	**DP**	1
EXP	Exposure	**DP**	**DP**	1
AEXP	AutoExposure	**DP**	**DP**	1
HUE	Hue	**DP**	**DP**	1
AFOCUS	AutoFocus	0 (Disable)	1 (Enable)	1
MFOCUS	Manuale Focus	**DP**	**DP**	1
LFREQ	Power Line Frequency 0 -> OFF 1 -> 50Hz 2 -> 60 Hz	0	2	1
GAIN	Gain	**DP**	**DP**	1
WBAL	White Balance	**DP**	**DP**	1
WAUTO	White Auto Balance	0 (Manual)	1 (Auto)	1
LEDON	Led On (Bit Mapped)	0	255	1
LEDPWR	Power LED %	0	100	1
LEDDEL	Delay Led ON before acq Image Unit Milliseconds	0	100000	1
IMGADD	Image Add	1	10	1

1.26.1 Read Par

Read Parameters **CMD=22 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	22	0

Data Received 97 (Length included)
Len=93

Byte 0-3	Byte 4	Byte 5-8	Byte 9-12	Byte 13-16	Byte 17-20	Byte 21-24	Byte 25-28
Len	22/0	SCALE	RANGLE	RCX	RCY	BRIGHT	SAT
Byte 29-32	Byte 33-36	Byte 37-40	Byte 41-44	Byte 45-48	Byte 49-52	Byte 53-56	Byte 57-60
CONT	SHARP	GAMMA	BACK	EXP	AEXP	HUE	AFOCUS
Byte 61-64	Byte 65-68	Byte 69-72	Byte 73-76	Byte 77-80	Byte 81-84	Byte 85-88	Byte 89-92
MFOCUS	LFREQ	GAIN	WBAL	WAUTO	LEDON	LEDPWR	LEDDEL
Byte 93-96							
IMGADD							

1.26.2 Set Par

Write Parameters **CMD=22 CMD_EXT=1**

Data Sent 98
Len=94

Byte 0-3	Byte 4	Byte 5	Byte 6-9	Byte 10-13	Byte 14-17	Byte 18-21	Byte 22-25
Len	22	1	SCALE	RANGLE	RCX	RCY	BRIGHT
Byte 26-29	Byte 30-33	Byte 34-37	Byte 38-41	Byte 42-45	Byte 46-49	Byte 50-53	Byte 54-57
SAT	CONT	SHARP	GAMMA	BACK	EXP	AEXP	HUE
Byte 58-61	Byte 62-65	Byte 66-69	Byte 70-73	Byte 74-77	Byte 78-81	Byte 82-85	Byte 86-89
AFOCUS	MFOCUS	LFREQ	GAIN	WBAL	WAUTO	LEDON	LEDPWR
Byte 90-93	Byte 94-97						
LEDDEL	IMGADD						

Data Received

Byte 0-3	Byte 4
1	22/0

1.26.3 Get Image

Get Data **CMD=22 CMD_EXT=2**

Data Sent

Byte 0-3	Byte 4	Byte 5	Byte 6 Send Image in UDP
3	22	2	0(not send) 1(send) 2(send data)

Data Received 4 (Length included)
Len=1

Byte 0-3	Byte 4
Len	22/0

1.27 Command Jobs

Command for JOBS

Parameters

TCP/IP	PxVisionBrowser	Val Min	Val Max	Ratio
JOBSNR	Number of Jobs files detected	0	xx	1
JOBNAME n	Job Name	string		1
JOBSET	Jib Name set for save or setting	string		1

1.27.1 Read Jobs

Read Parameters **CMD=23 CMD_EXT=0**

Data Sent

Byte 0-3	Byte 4	Byte 5
2	23	0

Data Received 9 + JOBSNR*JOBNAME(n).Lenght (Length included)
Len=5 + JOBSNR*JOBNAME(n).Lenght

Byte 0-3	Byte 4	Byte 5-8	Byte 9	Byte nn
Len	23/0	JOBSNR	JOBNAME n (0 end of string)	JOBNAME n +1 (0 end of string)

1.27.2 Save Job

Write Parameters **CMD=23 CMD_EXT=1**

Data Sent 6 + JOBSET.Lenght (Length included)
Len=2 + JOBSET.Lenght

Byte 0-3	Byte 4	Byte 5	Byte 6
Len	23	1	JOBSET (0 end of string)

Data Received

Byte 0-3	Byte 4
1	23/0

1.27.3 Set Job

Get Data **CMD=23 CMD_EXT=2**

Data Sent 6 + JOBSET.Lenght (Length included)
Len=2 + JOBSET.Lenght

Byte 0-3	Byte 4	Byte 5	Byte 6
Len	23	2	JOBSET (0 end of string)

Data Received

Byte 0-3	Byte 4
1	23/0

2 DLL Library PxEthClientLib

The **PxEthClientLib.dll** integrates all functions above explained, recallable from easy methods.
The Library is usable in .NET applications, some examples area downloadable from this link
Usage Example:

```
PxEthClientLib.PxEthClient MyClient;
MyClient = new PxEthClientLib.PxEthClient();
```

After is necessary to start the server by run **PxVisionBrowser** and make the connection:

```
PxEthClientLib.PxEthClient.PxEthClientResult  _Res= MyClient.Connect(IpAddr, 8500, 8501);
if (_Res == PxEthClientLib.PxEthClient.PxEthClientResult.ConnectionOk)
{
        // Connection Ok
}
else
{
        // Connection Error
}
```

If the connection is ok, the library is ready to use.
For Use the Library start the server from PxVisionBrowser
PxVisionBrowser User Guide

2.1 PxEthClient

Following are explained all methods of class **PxEthClient**

2.1.1 Connect

Connection to server of **PxVisionBrowser**

public PxEthClientResult **Connect**(string **IpAddr**, int **CommandPort**, int **VideoPort**)

Parameters

String **IpAddr**	Server **IP** Address of **PxVisionBrowser**
Int **CommandPort**	Command Port (8500)
Int **VideoPort**	**UDP** Port (8501)

Returned

PxEthClientResult.ConnectionOk	Connessione Ok
PxEthClientResult.VideoPortConnectionError	Video Port connection Error
PxEthClientResult.CommandPortConnectionError	Command port connection Error

---

2.1.2 Disconnect

Effettua la disconnessione al server di **PxVisionBrowser**

public void **Disconnect**()

Parameters

None

Returned

void

2.2 BarCodeReader

Following are explained all methods of class **BarCodeReader** in the namespace **PxBarCodeReader**
This static class use two classes, BarCodeReaderPar.cs and BarCodeReaderDataGet.cs

Public Variables PxBarCodeReader.BarCodeReaderPar

---

public int **CameraRegionX**
public int **CameraRegionY**
public int **CameraRegionWidth**
public int **CameraRegionHeight**
public int **BorderLeft**
public int **BorderRight**
public int **BorderTop**
public int **BorderBottom**
public byte **BorderColorRed**
public byte **BorderColorGreen**
public byte **BorderColorBlue**
public byte **AutoRotate**

Public Variables PxBarCodeReader.BarCodeReaderDataGet

---

public bool **AcqOk**
public string **BarCodeResult**
public string **BarCodeFormat**

METODI PxBarCodeReader.BarCodeReader

---

public static PxEthClient.PxEthClientResult **SetPar**(BarCodeReaderPar **Par**)
Set the function parameters

Parameters

BarCodeReaderPar **Par**	Function Parameters

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out BarCodeReaderPar **Par**)
Function Parameters Read

Parameters

out BarCodeReaderPar **Par**	Parameters Read

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out BarCodeReaderDataGet **Data**)
public static PxEthClient.PxEthClientResult **Get**(out BarCodeReaderDataGet **Data**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function BarCode

Parameters

out BarCodeReaderDataGet **Data**	Data Read
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.3 BrightnessDetector

Following are explained all methods of class **BrightnessDetector** in the namespace **PxDetector**
This static class use two classes, BrightnessDetectorPar.cs and BrightnessDetectorDataGet.cs

Public Variables PxDetector.BrightnessDetectorPar

---

public int[] **ParRegion**
Array of 4 integer for region used based on type
TypeRegion=0 (rectangle)
ParRegion[0] X
ParRegion[1] Y
ParRegion[2] Width
ParRegion[3] Heght
TypeRegion=1 (Circle)
ParRegion[0] Center X
ParRegion[1] Center Y
ParRegion[2] Radius
ParRegion[3] not used
public byte **TypeRegion**
Type of Region
TypeRegion=0 (rectangle)
TypeRegion=1 (Circle)
public byte **ThMin**
public byte **ThMax**
public byte **RGBchannel**

Public Variables PxDetector.BrightnessDetectorDataGet

---

public bool **AcqOk**

METHODS PxDetector.BrightnessDetector

---

public static PxEthClient.PxEthClientResult **SetPar**(BrightnessDetectorPar[] **Par**)
Set the function parameters

Parameters

BrightnessDetectorPar[] **Par**	Array of Parameters based on Objects number configured

Returned

PxEthClient.PxEthClientResult.NoObjectsConfigured	None object configured Par=null or length =0
PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out BrightnessDetectorPar[] **Par**)
Function Parameters Read

Parameters

out BrightnessDetectorPar[] **Par**	Parameters for the detector read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out BrightnessDetectorDataGet[] **Data**)
public static PxEthClient.PxEthClientResult **Get**(out BrightnessDetectorDataGet[] **Data**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Detector

Parameters

out BrightnessDetectorDataGet[] **Data**	Data Read for all detector Configured
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.NoObjectsConfigured	No Objects Configured
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.4 Calliper

Following are explained all methods of class **Callifor the**n the namespace **PxMeasure**
This static class use two classes, CalliperPar.cs and CalliperDataGet.cs

Public Variables PxMeasure.CalliperPar

---

public int **StartX**
public int **StartY**
public int **EndX**
public int **EndY**
public int **Width**
public byte **Inside**
public byte **PixelNoise**
public byte **PixelPressureHigh**
public byte **PixelPressureLow**
public byte **Dark**
public byte **GrayValue**

Public Variables PxMeasure.CalliperDataGet

---

public bool **AcqOk**
public int **Measure**
public int **BaseXintersectionPoint**
public int **BaseYintersectionPoint**
public int **OpeningXintersectionPoint**
public int **OpeningYintersectionPoint**

METHODS PxMeasure.Calliper

---

public static PxEthClient.PxEthClientResult **SetPar**(CalliperPar[] **Par**)
Set the function parameters

Parameters

CalliperPar[] **Par**	Array of Parameters based on Objects number configured

Returned

PxEthClient.PxEthClientResult.NoObjectsConfigured	None object configured Par=null or length =0
PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out CalliperPar[] **Par**)
Function Parameters Read

Parameters

out CalliperPar[] **Par**	Parameters for objects read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out CalliperDataGet[] **Data**, out double **ScaleFactor**)
public static PxEthClient.PxEthClientResult **Get**(out CalliperDataGet[] **Data**, out double **ScaleFactor**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Detector

Parameters

out CalliperDataGet [] **Data**	Data Read for all objects Configured
out double **ScaleFactor**	Scale Factor Pixel/mm
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.NoObjectsConfigured	No Objects Configured
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.5 Camera Setting

Following are explained all methods of class **CameraSetting** in the namespace **PxCameraSetting**
This static class use a class, CameraSettingPar.cs

Public Variables PxCameraSetting.CameraSettingPar

---

public double **ConversionFactorMM**
public double **CameraRotationAngle**
public int **CameraRotationXcenter**
public int **CameraRotationYcenter**
public int **BrighTness**
public int **Saturation**
public int **Contrast**
public int **SharpNess**
public int **Gamma**
public int **BackLight**
public int **Exposure**
public int **AutoExposure**
public int **Hue**
public int **AutoFocus**
public int **ManualFocus**
public int **PowerLineFrequency**
public int **Gain**
public int **WhiteBalance**
public int **WhiteAuto**
public int **LedOn**
public int **PowerLed**
public int **DelayLedOn**
public int **ImagesAdd**

METHODS PxCameraSetting.CameraSetting

---

public static PxEthClient.PxEthClientResult **SetPar**(CameraSettingPar **Par**)
Set the function parameters

Parameters

CameraSettingPar **Par**	Camera Setting Par

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out CameraSettingPar **Par**)
Function Parameters Read

Parameters

out CameraSettingPar **Par**	Parameters Read

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **GetImage**(out ImageSource **CameraImage**)
Legge una frame dalla camera

Parameters

out ImageSource **CameraImage**	Image Capture from camera

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.6 CircleMarker

Following are explained all methods of class **CircleMarker** in the namespace **PxMarkerDetection**
This static class use two classes, MarkerPar.cs and MarkerDataGet.cs

Public Variables PxMarkerDetection.MarkerPar

---

public int **CameraRegionX**
public int **CameraRegionY**
public int **CameraRegionWidth**
public int **CameraRegionHeight**
public int **BorderLeft**
public int **BorderRight**
public int **BorderTop**
public int **BorderBottom**
public int **Diameter**
public byte **BorderColorRed**
public byte **BorderColorGreen**
public byte **BorderColorBlue**
public double **Accuracy**
public byte **Tolerance**
public byte **PercentageDiameter**
public byte **PixelNoise**
public byte **PixelPressure**
public byte **BlackValue**
public byte **WhiteValue**

Public Variables PxMarkerDetection.MarkerDataGet

---

public bool **AcqOk**
public double **ScaleFactor**
public int **CenterXrefCameraCenter**
public int **CenterYrefCameraCenter**
public int **CenterXrefCameraOrigin**
public int **CenterYrefCameraOrigin**
public double **Angle**
public int **Width**
public int **Height**
public int[] **BoundingPoints**

METHODS PxMarkerDetection.CircleMarker

---

public static PxEthClient.PxEthClientResult **SetPar**(MarkerPar **Par**)
Set the function parameters

Parameters

MarkerPar **Par**	Parameters Marker

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out MarkerPar **Par**)
Function Parameters Read

Parameters

out MarkerPar **Par**	Parameters per marker read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out MarkerDataGet **Data**)
public static PxEthClient.PxEthClientResult **Get**(out MarkerDataGet **Data**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Marker

Parameters

out MarkerDataGet **Data**	Data Read for the marker
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.7 ContrastDetector

Following are explained all methods of class **ContrastDetector** in the namespace **PxDetector**
This static class use two classes, ContrastDetectorPar.cs and ContrastDetectorDataGet.cs

Public Variables PxDetector.ContrastDetectorPar

---

public int[] **ParRegion**
Array of 4 integer for region used based on type
Tyforegion=0 (rectangle)
ParRegion[0] X
ParRegion[1] Y
ParRegion[2] Width
ParRegion[3] Heght
Tyforegion=1 (Circle)
ParRegion[0] Center X
ParRegion[1] Center Y
ParRegion[2] Radius
ParRegion[3] not used
public byte **Tyforegion**
Type of Region
Tyforegion=0 (rectangle)
Tyforegion=1 (Circle)
public byte **ThMin**
public byte **ThMax**
public byte **RGBchannel**

Public Variables PxDetector.ContrastDetectorDataGet

---

public bool **AcqOk**

METHODS PxDetector.BrightnessDetector

---

public static PxEthClient.PxEthClientResult **SetPar**(ContrastDetectorPar[] **Par**)
Set the function parameters

Parameters

ContrastDetectorPar[] **Par**	Array of Parameters based on Objects number configured

Returned

PxEthClient.PxEthClientResult.NoObjectsConfigured	None object configured Par=null or length =0
PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out ContrastDetectorPar[] **Par**)
Function Parameters Read

Parameters

out ContrastDetectorPar[] **Par**	Parameters for the detector read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out ContrastDetectorDataGet[] **Data**)
public static PxEthClient.PxEthClientResult **Get**(out ContrastDetectorDataGet[] **Data**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Detector

Parameters

out ContrastDetectorDataGet[] **Data**	Data Read for all detector Configured
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.NoObjectsConfigured	No Objects Configured
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.8 CrossSection

Following are explained all methods of class **CrossSection** in the namespace **PxMeasure**
This static class use two classes, CrossSectionPar.cs and CrossSectionDataGet.cs

Public Variables PxMeasure.CrossSectionPar

---

public int **StartX**
public int **StartY**
public int **EndX**
public int **EndY**
public byte **Inside**
public byte **PixelNoise**
public byte **PixelPressureHigh**
public byte **PixelPressureLow**
public byte **Dark**
public byte **GrayValue**

Public Variables PxMeasure.CrossSectionDataGet

---

public bool **AcqOk**
public int **Measure**
public int **BaseXintersectionPoint**
public int **BaseYintersectionPoint**
public int **OpeningXintersectionPoint**
public int **OpeningYintersectionPoint**

METHODS PxMeasure.Callifor

---

public static PxEthClient.PxEthClientResult **SetPar**(CrossSectionPar[] **Par**)
Set the function parameters

Parameters

CrossSectionPar[] **Par**	Array of Parameters based on Objects number configured.

Returned

PxEthClient.PxEthClientResult.NoObjectsConfigured	None object configured Par=null or length =0
PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out CrossSectionPar[] **Par**)
Function Parameters Read

Parameters

out CrossSectionPar[] **Par**	Parameters for objects read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out CrossSectionDataGet[] **Data**, out double **ScaleFactor**)
public static PxEthClient.PxEthClientResult **Get**(out CrossSectionDataGet[] **Data**, out double **ScaleFactor**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Detector

Parameters

out CrossSectionDataGet[] **Data**	Data Read for all objects Configured
out double **ScaleFactor**	Scale Factor Pixel/mm
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.NoObjectsConfigured	No Objects Configured
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.9 Filtering

Following are explained all methods of class **Filtering** in the namespace **PxEthClientGeneral**
This static class use a class, FilterPar.cs

Public Variables PxEthClientGeneral.FilterPar

---

public int **Gaussian**
public int **Blur**
public int **MedianBlur**
public int **Laplacian**
public int **Erode**
public int **Dilate**
public double **Gamma**
public int **GrayImage**
public int **ThresHoldType**
public int **ThreSholdValue**
public int **ThreSholdMaxValue**
public int **AdaptiveParam1**
public int **CannyThreShold1**
public int **CannyThreShold2**
public double **Alpha**
public int **Beta**
public byte[] **FilterPosition**

METHODS PxEthClientGeneral.Filtering

---

public static PxEthClient.PxEthClientResult **SetPar**(FilterPar **Par**)
Set the function parameters

Parameters

FilterPar **Par**	Filter Par

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out FilterPar **Par**)
Function Parameters Read

Parameters

out FilterPar **Par**	Parameters Read

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.10 FindMatchMarker

Following are explained all methods of class **FindMatchMarker** in the namespace **PxMarkerDetection**
This static class use two classes, FindMatchPar.cs and MarkerDataGet.cs

Public Variables PxMarkerDetection.FindMatchPar

---

public int **CameraRegionX**
public int **CameraRegionY**
public int **CameraRegionWidth**
public int **CameraRegionHeight**
public int **RegionWidthOk**
public int **RegionHeightOk**
public int **ScoreOk**
public byte **DealtaScoreOk**
public byte **DealtaRegionOk**
public string **ModelImage**

Public Variables PxMarkerDetection.MarkerDataGet

---

public bool **AcqOk**
public double **ScaleFactor**
public int **CenterXrefCameraCenter**
public int **CenterYrefCameraCenter**
public int **CenterXrefCameraOrigin**
public int **CenterYrefCameraOrigin**
public double **Angle**
public int **Width**
public int **Height**
public int[] **BoundingPoints**

METHODS PxMarkerDetection.FindMatchMarker

---

public static PxEthClient.PxEthClientResult **SetPar**(FindMatchPar **Par**)
Set the function parameters

Parameters

FindMatchPar **Par**	Parameters Marker

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out FindMatchPar **Par**)
Function Parameters Read

Parameters

out FindMatchPar **Par**	Parameters for marker read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out MarkerDataGet **Data**)
public static PxEthClient.PxEthClientResult **Get**(out MarkerDataGet **Data**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Marker

Parameters

out MarkerDataGet **Data**	Data Read for the marker
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.11 FitCircle

Following are explained all methods of class **FitCircle** in the namespace **PxMeasure**
This static class use two classes, FitCirclePar.cs and FitCircleDataGet.cs

Public Variables PxMeasure.FitCirclePar

---

public int **CenterX**
public int **CenterY**
public int **ExternalRadius**
public int **InternalRadius**
public int **Width**
public double **StartAngle**
public double **EndAngle**
public double **StepAngle**
public byte **AcqDirection**
public byte **PixelNoise**
public byte **PixelPressure**
public byte **Dark**
public byte **GrayValue**
public byte **TypeMethod**

Public Variables PxMeasure.FitCircleDataGet

---

public bool **AcqOk**
public int **FitCircleStartX**
public int **FitCircleStartY**
public int **FitCircleEndX**
public int **FitCircleEndY**
public double **FitCircleRadius**
public int **FitCircleCenterX**
public int **FitCircleCenterY**

METHODS PxMeasure.FitCircle

---

public static PxEthClient.PxEthClientResult **SetPar**(FitCirclePar[] **Par**)
Set the function parameters

Parameters

FitCirclePar[] **Par**	Array of Parameters based on Objects number configured.

Returned

PxEthClient.PxEthClientResult.NoObjectsConfigured	None object configured Par=null or length =0
PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out FitCirclePar[] **Par**)
Function Parameters Read

Parameters

out FitCirclePar[] **Par**	Parameters for objects read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out FitCircleDataGet[] **Data**, out double **ScaleFactor**)
public static PxEthClient.PxEthClientResult **Get**(out FitCircleDataGet[] **Data**, out double **ScaleFactor**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Detector

Parameters

out FitCircleDataGet[] **Data**	Data Read for all objects Configured
out double **ScaleFactor**	Scale Factor Pixel/mm
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.NoObjectsConfigured	No Objects Configured
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.12 FitLine

Following are explained all methods of class **FitLine** in the namespace **PxMeasure**
This static class use two classes, FitLinePar.cs and FitLineDataGet.cs

Public Variables PxMeasure.FitLinePar

---

public int **StartX**
public int **StartY**
public int **EndX**
public int **EndY**
public int **Width**
public int **Step**
public byte **AcqDirection**
public byte **PixelNoise**
public byte **PixelPressure**
public byte **Dark**
public byte **GrayValue**
public byte **TypeMethod**

Public Variables PxMeasure.FitLineGet

---

public bool **AcqOk**
public int **FitLineStartX**
public int **FitLineStartY**
public int **FitLineEndX**
public int **FitLineEndY**
public double **FitLineAngle**
public int **FitLineLenght**

METHODS PxMeasure.FitLine

---

public static PxEthClient.PxEthClientResult **SetPar**(FitLine[] **Par**)
Set the function parameters

Parameters

FitLine[] **Par**	Array of Parameters based on Objects number configured

Returned

PxEthClient.PxEthClientResult.NoObjectsConfigured	None object configured Par=null or length =0
PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out FitLine[] **Par**)
Function Parameters Read

Parameters

out FitLine[] **Par**	Parameters for objects read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out FitLineDataGet[] **Data**, out double **ScaleFactor**)
public static PxEthClient.PxEthClientResult **Get**(out FitLineDataGet[] **Data**, out double **ScaleFactor**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Detector

Parameters

out FitLineDataGet[] **Data**	Data Read for all objects Configured
out double **ScaleFactor**	Scale Factor Pixel/mm
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.NoObjectsConfigured	No Objects Configured
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.13 GapCrossSection

Following are explained all methods of class **GapCrossSection** in the namespace **PxMeasure**
This static class use two classes, GapCrossSectionPar.cs and GapCrossSectionDataGet.cs

Public Variables PxMeasure.GapCrossSectionPar

---

public int **StartX**
public int **StartY**
public int **EndX**
public int **EndY**
public int **Width**
public int **GapMinValue**
public byte **Inside**
public byte **PixelNoise**
public byte **PixelPressureHigh**
public byte **PixelPressureLow**
public byte **Dark**
public byte **GrayValue**

Public Variables PxMeasure.GapCrossSectionDataGet

---

public bool **AcqOk**
public int **MeasureMin**
public int **MeasureMax**
public int **MinBaseXintersectionPoint**
public int **MinBaseYintersectionPoint**
public int **MinOpeningXintersectionPoint**
public int **MinOpeningYintersectionPoint**
public int **MaxBaseXintersectionPoint**
public int **MaxBaseYintersectionPoint**
public int **MaxOpeningXintersectionPoint**
public int **MaxOpeningYintersectionPoint**

METHODS PxMeasure.FitLine

---

public static PxEthClient.PxEthClientResult **SetPar**(GapCrossSectionPar[] **Par**)
Set the function parameters

Parameters

GapCrossSectionPar[] **Par**	Array of Parameters based on Objects number configured

Returned

PxEthClient.PxEthClientResult.NoObjectsConfigured	None object configured Par=null or length =0
PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out GapCrossSectionPar[] **Par**)
Function Parameters Read

Parameters

out GapCrossSectionPar[] **Par**	Parameters for objects read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out GapCrossSectionDataGet[] **Data**, out double **ScaleFactor**)
public static PxEthClient.PxEthClientResult **Get**(out GapCrossSectionDataGet[] **Data**, out double **ScaleFactor**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Detector

Parameters

out GapCrossSectionDataGet[] **Data**	Data Read for all objects Configured
out double **ScaleFactor**	Scale Factor Pixel/mm
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.NoObjectsConfigured	No Objects Configured
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.14 GrayDetector

Following are explained all methods of class **GrayDetector** in the namespace **PxDetector**
This static class use two classes, GrayDetectorPar.cs and GrayDetectorDataGet.cs

Public Variables PxDetector.GrayDetectorPar

---

public int[] **ParRegion**
Array of 4 integer for region used based on type
Tyforegion=0 (rectangle)
ParRegion[0] X
ParRegion[1] Y
ParRegion[2] Width
ParRegion[3] Heght
Tyforegion=1 (Circle)
ParRegion[0] Center X
ParRegion[1] Center Y
ParRegion[2] Radius
ParRegion[3] not used
public byte **Tyforegion**
Type of Region
Tyforegion=0 (rectangle)
Tyforegion=1 (Circle)
public byte **GrayMinValue**
public byte **GrayMaxValue**
public byte **ThMin**
public byte **ThMax**
public byte **RGBchannel**

Public Variables PxDetector.GrayDetectorDataGet

---

public bool **AcqOk**

METHODS PxDetector.GrayDetector

---

public static PxEthClient.PxEthClientResult **SetPar**(GrayDetectorPar[] **Par**)
Set the function parameters

Parameters

GrayDetectorPar[] **Par**	Array of Parameters based on Objects number configured

Returned

PxEthClient.PxEthClientResult.NoObjectsConfigured	None object configured Par=null or length =0
PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out GrayDetectorPar[] **Par**)
Function Parameters Read

Parameters

out GrayDetectorPar[] **Par**	Parameters for the detector read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out GrayDetectorDataGet[] **Data**)
public static PxEthClient.PxEthClientResult **Get**(out GrayDetectorDataGet[] **Data**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Detector

Parameters

out ContrastDetectorDataGet[] **Data**	Data Read for all detector Configured
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.NoObjectsConfigured	No Objects Configured
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.15 HoleMarker

Following are explained all methods of class **HoleMarker** in the namespace **PxMarkerDetection**
This static class use two classes, HolePar.cs and MarkerDataGet.cs

Public Variables PxMarkerDetection.HolePar

---

public int **CameraRegionX**
public int **CameraRegionY**
public int **CameraRegionWidth**
public int **CameraRegionHeight**
public int **BorderLeft**
public int **BorderRight**
public int **BorderTop**
public int **BorderBottom**
public int **HoleDiameter**
public byte **BorderColorRed**
public byte **BorderColorGreen**
public byte **BorderColorBlue**
public byte **Tolerance**

Public Variables PxMarkerDetection.MarkerDataGet

---

public bool **AcqOk**
public double **ScaleFactor**
public int **CenterXrefCameraCenter**
public int **CenterYrefCameraCenter**
public int **CenterXrefCameraOrigin**
public int **CenterYrefCameraOrigin**
public double **Angle**
public int **Width**
public int **Height**
public int[] **BoundingPoints**

METHODS PxMarkerDetection.HoleMarker

---

public static PxEthClient.PxEthClientResult **SetPar**(HolePar **Par**)
Set the function parameters

Parameters

HolePar **Par**	Parameters Marker

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out HolePar **Par**)
Function Parameters Read

Parameters

out HolePar **Par**	Parameters for marker read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out MarkerDataGet **Data**)
public static PxEthClient.PxEthClientResult **Get**(out MarkerDataGet **Data**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Marker

Parameters

out MarkerDataGet **Data**	Data Read for the marker
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.16 ImageMask

Following are explained all methods of class **ImageMask** in the namespace **PxImageMask**
This static class use a class, ImageMaskPar.cs

Public Variables PxImageMask.ImageMaskPar

---

public int **CameraRegionX**
public int **CameraRegionY**
public int **CameraRegionWidth**
public int **CameraRegionHeight**
public byte **ImageMaskEnable**
public byte **UseImageMaskLastFrame**
public byte **ImageMaskMode**
public string **MaskImage**

METHODS PxImageMask.ImageMask

---

public static PxEthClient.PxEthClientResult **SetPar**(ImageMaskPar **Par**)
Set the function parameters

Parameters

ImageMaskPar **Par**	Parameters ImageMask

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out ImageMaskPar **Par**)
Function Parameters Read

Parameters

out ImageMaskPar **Par**	Parameters for the ImageMask read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.17 JobsManager

Following are explained all methods of class **JobsManager** in the namespace **PxJobsManager**

METHODS PxJobsManager.JobsManager

---

public static PxEthClient.PxEthClientResult **SaveJobName**(string **JobName**)
Save a Job

Parameters

string **JobName**	Job File Name

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **SaveJobDefault**()
Save a default "PxVision.cfg"

Parameters

None

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadJobs** (out string[] **Jobs**)
Read all Jobs Configured

Parameters

Out string[] **Jobs**	Array of string name Jobs Configured

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **SetJobDefault** ()
Set "PxVision.cfg" as current Job

Parameters

None

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

public static PxEthClient.PxEthClientResult **SetJobName**(string **JobName**)
Set a Job file

Parameters

string **JobName**	Job name

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.18 ObjectsAreaDetector

Following are explained all methods of class **ObjectsAreaDetector** in the namespace **PxDetector**
This static class use three classes, ObjectsAreaDetectorPar.cs , ObjectsAreaDetectorDataGet.cs and ObjectsDetected.cs

Public Variables PxDetector.ObjectsAreaDetectorPar

---

public int **RegionX**
public int **RegionY**
public int **RegionWidth**
public int **RegionHeight**
public int **LeftBorder**
public int **RightBorder**
public int **TopBorder**
public int **BottomBorder**
public int **MinAreaDetected**
public int **MaxAreaDetected**
public int **RectWidthDetected**
public int **RectHeightDetected**
public int **CircleDiameterDetected**
public byte **BlueBorder**
public byte **GreenBorder**
public byte **RedBorder**
public byte **RectWidthTolerance**
public byte **RectHeightolerance**
public byte **DiameterTolerance**
public byte **AreaType**

Public Variables PxDetector.ObjectsAreaDetectorDataGet

---

public List<ObjectsDetected> **ObjectsDetected**

Public Variables PxDetector.ObjectsDetected

---

public int **BoundingX**
public int **BoundingY**
public int **BoundingWidth**
public int **BoundingHeight**
public int[] **MinRectPoints**
public int **CenterXrefCameraCenter**
public int **CenterYrefCameraCenter**
public int **CenterXrefCameraOrigin**
public int **CenterYrefCameraOrigin**
public int **AreaDetected**
public double **Angle**

METHODS PxDetector.ObjectsAreaDetector

---

public static PxEthClient.PxEthClientResult **SetPar**(ObjectsAreaDetectorPar[] **Par**)
Set the function parameters

Parameters

ObjectsAreaDetectorPar[] **Par**	Array of Parameters based on Objects number configured

Returned

PxEthClient.PxEthClientResult.NoObjectsConfigured	None object configured Par=null or length =0
PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out ObjectsAreaDetectorPar[] **Par**)
Function Parameters Read

Parameters

out ObjectsAreaDetectorPar[] **Par**	Parameters for the detector read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out ObjectsAreaDetectorDataGet[] **Data** , out double **ScaleFactor**)
public static PxEthClient.PxEthClientResult **Get**(out ObjectsAreaDetectorDataGet[] **Data**, , out double **ScaleFactor**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Detector

Parameters

out ObjectsAreaDetectorDataGet[] **Data**	Data Read for all detector Configured
out double **ScaleFactor**	Fattore di scala Pixel/mm
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.NoObjectsConfigured	No Objects Configured
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.19 ObjectsColorDetector

Following are explained all methods of class **ObjectsColorDetector** in the namespace **PxDetector**
This static class use three classes, ObjectsColorDetectorPar.cs , ObjectsColorDetectorDataGet.cs and ObjectsDetected.cs

Public Variables PxDetector.ObjectsColorDetectorPar

---

public int **RegionX**
public int **RegionY**
public int **RegionWidth**
public int **RegionHeight**
public int **LeftBorder**
public int **RightBorder**
public int **TopBorder**
public int **BottomBorder**
public int **MinAreaDetected**
public int **MaxAreaDetected**
public int **RectWidthDetected**
public int **RectHeightDetected**
public int **CircleDiameterDetected**
public int **HminValue**
public int **HmaxValue**
public byte **SminValue**
public byte **SmaxValue**
public byte **VminValue**
public byte **VmaxValue**
public byte **BlueBorder**
public byte **GreenBorder**
public byte **RedBorder**
public byte **RectWidthTolerance**
public byte **RectHeightolerance**
public byte **DiameterTolerance**
public byte **AreaType**

Public Variables PxDetector.ObjectsColorDetectorDataGet

---

public List<ObjectsDetected> **ObjectsDetected**

Public Variables PxDetector.ObjectsDetected

---

public int **BoundingX**
public int **BoundingY**
public int **BoundingWidth**
public int **BoundingHeight**
public int[] **MinRectPoints**
public int **CenterXrefCameraCenter**
public int **CenterYrefCameraCenter**
public int **CenterXrefCameraOrigin**
public int **CenterYrefCameraOrigin**
public int **AreaDetected**
public double **Angle**

METHODS PxDetector.ObjectsColorDetector

---

public static PxEthClient.PxEthClientResult **SetPar**(ObjectsColorDetectorPar[] **Par**)
Set the function parameters

Parameters

ObjectsColorDetectorPar[] **Par**	Array of Parameters based on Objects number configured

Returned

PxEthClient.PxEthClientResult.NoObjectsConfigured	None object configured Par=null or length =0
PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out ObjectsColorDetectorPar[] **Par**)
Function Parameters Read

Parameters

out ObjectsColorDetectorPar[] **Par**	Parameters for the detector read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out ObjectsColorDetectorDataGet[] **Data** , out double **ScaleFactor**)
public static PxEthClient.PxEthClientResult **Get**(out ObjectsColorDetectorDataGet[] **Data**, , out double **ScaleFactor**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Detector

Parameters

out ObjectsColorDetectorDataGet[] **Data**	Data Read for all detector Configured
out double **ScaleFactor**	Fattore di scala Pixel/mm
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.NoObjectsConfigured	No Objects Configured
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.20 OcrReader

Following are explained all methods of class **OcrReader** in the namespace **PxOcrReader**
This static class use two classes, OcrReaderPar.cs and OcrReaderDataGet.cs

Public Variables PxOcrReader.OcrReaderPar

---

public int **CameraRegionX**
public int **CameraRegionY**
public int **CameraRegionWidth**
public int **CameraRegionHeight**
public int **BorderLeft**
public int **BorderRight**
public int **BorderTop**
public int **BorderBottom**
public byte **BorderColorRed**
public byte **BorderColorGreen**
public byte **BorderColorBlue**
public byte **Engine**
public byte **ScoreOk**
public string **Language**

Public Variables PxOcrReader.OcrReaderDataGet

---

public bool **AcqOk**
public byte **Score**
public string **OcrResult**

METHODS PxOcrReader.OcrReader

---

public static PxEthClient.PxEthClientResult **SetPar**(OcrReaderPar **Par**)
Set the function parameters

Parameters

OcrReaderPar **Par**	Function Parameters

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out OcrReaderPar **Par**)
Function Parameters Read

Parameters

out OcrReaderPar **Par**	Parameters Read

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out OcrReaderDataGet **Data**)
public static PxEthClient.PxEthClientResult **Get**(out OcrReaderDataGet **Data**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function BarCode

Parameters

out OcrReaderDataGet **Data**	Data Read
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.21 ProbeLine

Following are explained all methods of class **ProbeLine** in the namespace **PxProbe**
This static class use two classes, ProbeLinePar.cs and ProbeLineDataGet.cs

Public Variables PxProbe.ProbeLinePar

---

public int **StartX**
public int **StartY**
public int **EndX**
public int **EndY**
public int **Width**
public byte **PixelNoise**
public byte **PixelPressure**
public byte **Dark**
public byte **GrayValue**

Public Variables PxProbe.ProbeLineDataGet

---

public bool **AcqOk**
public int **CenterXrefCameraCenter**
public int **CenterYrefCameraCenter**
public int **CenterXrefCameraOrigin**
public int **CenterYrefCameraOrigin**

METHODS PxProbe.ProbeLine

---

public static PxEthClient.PxEthClientResult **SetPar**(ProbeLinePar **Par**)
Set the function parameters

Parameters

ProbeLinePar **Par**	Function Parameters

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out ProbeLinePar **Par**)
Function Parameters Read

Parameters

out ProbeLinePar **Par**	Parameters Read

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out ProbeLineDataGet **Data**, out double **ScaleFactor**)
public static PxEthClient.PxEthClientResult **Get**(out ProbeLineDataGet **Data**, out double **ScaleFactor**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function BarCode

Parameters

out ProbeLineDataGet **Data**	Data Read
out double **ScaleFactor**	Fattore di scala Pixel/mm
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.22 ProbeSinglePoint

Following are explained all methods of class **ProbeSinglePoint** in the namespace **PxProbe**
This static class use two classes, ProbeSinglePointPar.cs and ProbeSinglePointDataGet.cs

Public Variables PxProbe.ProbeSinglePointPar

---

public int **StartX**
public int **StartY**
public int **EndX**
public int **EndY**
public byte **PixelNoise**
public byte **PixelPressure**
public byte **Dark**
public byte **GrayValue**

Public Variables PxProbe.ProbeSinglePointDataGet

---

public bool **AcqOk**
public int **CenterXrefCameraCenter**
public int **CenterYrefCameraCenter**
public int **CenterXrefCameraOrigin**
public int **CenterYrefCameraOrigin**

METHODS PxProbe.ProbeSinglePoint

---

public static PxEthClient.PxEthClientResult **SetPar**(ProbeSinglePointPar **Par**)
Set the function parameters

Parameters

ProbeSinglePointPar **Par**	Function Parameters

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out ProbeSinglePointPar **Par**)
Function Parameters Read

Parameters

out ProbeSinglePointPar **Par**	Parameters Read

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out ProbeSinglePointDataGet **Data**, out double **ScaleFactor**)
public static PxEthClient.PxEthClientResult **Get**(out ProbeSinglePointDataGet **Data**, out double **ScaleFactor**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function BarCode

Parameters

out ProbeSinglePointDataGet **Data**	Data Read
out double **ScaleFactor**	Fattore di scala Pixel/mm
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.23 SquareMarker

Following are explained all methods of class **SquareMarker** in the namespace **PxMarkerDetection**
This static class use two classes, SquarePar.cs and MarkerDataGet.cs

Public Variables PxMarkerDetection.SquarePar

---

public int **CameraRegionX**
public int **CameraRegionY**
public int **CameraRegionWidth**
public int **CameraRegionHeight**
public int **BorderLeft**
public int **BorderRight**
public int **BorderTop**
public int **BorderBottom**
public int **SquareWidth**
public int **SquareHeight**
public double **Accuracy**
public byte **BorderColorRed**
public byte **BorderColorGreen**
public byte **BorderColorBlue**
public byte **Tolerance**
public byte **BoundingBox**

Public Variables PxMarkerDetection.MarkerDataGet

---

public bool **AcqOk**
public double **ScaleFactor**
public int **CenterXrefCameraCenter**
public int **CenterYrefCameraCenter**
public int **CenterXrefCameraOrigin**
public int **CenterYrefCameraOrigin**
public double **Angle**
public int **Width**
public int **Height**
public int[] **BoundingPoints**

METHODS PxMarkerDetection.SquareMarker

---

public static PxEthClient.PxEthClientResult **SetPar**(SquarePar **Par**)
Set the function parameters

Parameters

SquarePar **Par**	Parameters Marker

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out SquarePar **Par**)
Function Parameters Read

Parameters

out SquarePar **Par**	Parameters for marker read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out MarkerDataGet **Data**)
public static PxEthClient.PxEthClientResult **Get**(out MarkerDataGet **Data**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Marker

Parameters

out MarkerDataGet **Data**	Data Read for the marker
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

2.24 Type1Marker

Following are explained all methods of class **Type1Marker** in the namespace **PxMarkerDetection**
This static class use two classes, MarkerPar.cs and MarkerDataGet.cs

Public Variables PxMarkerDetection.MarkerPar

---

public int **CameraRegionX**
public int **CameraRegionY**
public int **CameraRegionWidth**
public int **CameraRegionHeight**
public int **BorderLeft**
public int **BorderRight**
public int **BorderTop**
public int **BorderBottom**
public int **Diameter**
public double **Accuracy**
public byte **BorderColorRed**
public byte **BorderColorGreen**
public byte **BorderColorBlue**
public byte **Tolerance**
public byte **ForcentageDiameter**
public byte **PixelNoise**
public byte **PixelPressure**
public byte **BlackValue**
public byte **WhiteValue**

Public Variables PxMarkerDetection.MarkerDataGet

---

public bool **AcqOk**
public double **ScaleFactor**
public int **CenterXrefCameraCenter**
public int **CenterYrefCameraCenter**
public int **CenterXrefCameraOrigin**
public int **CenterYrefCameraOrigin**
public double **Angle**
public int **Width**
public int **Height**
public int[] **BoundingPoints**

METHODS PxMarkerDetection.Type1Marker

---

public static PxEthClient.PxEthClientResult **SetPar**(MarkerPar **Par**)
Set the function parameters

Parameters

MarkerPar **Par**	Parameters Marker

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **ReadPar**(out MarkerPar **Par**)
Function Parameters Read

Parameters

out MarkerPar **Par**	Parameters for marker read for the function

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

---

public static PxEthClient.PxEthClientResult **Get**(out MarkerDataGet **Data**)
public static PxEthClient.PxEthClientResult **Get**(out MarkerDataGet **Data**, out ImageSource **CameraImage**, bool **PrintData**)
Read Data from function Marker

Parameters

out MarkerDataGet **Data**	Data Read for the marker
out ImageSource **CameraImage**	Image Capture from camera
bool **PrintData**	**True** The data are drawn in the Image CameraImage **False** None data drawn

Returned

PxEthClient.PxEthClientResult.SendError	Data Send Error
PxEthClient.PxEthClientResult.ReceiveError	Data Receive Error
PxEthClient.PxEthClientResult.CommandError	Command Error
PxEthClient.PxEthClientResult.CommandOk	Command Ok

Index