Programmer´s Guide for feature HDR
AN201414/v1.0/2015-02-06

Description

HDR (High Dynamic Range) is a technique that allows a greater dynamic range of luminance between the lightest and darkest areas of an image.

This document describes the programming of the HDR parameters.

Products

The following camera models support the HDR feature:

GigE	Dual-GigE	USB
VisiLine®	HXG (Rel. 2)	MXU
VLG-22M / VLG-22C VLG-40M / VLG-40C	all models	MXUC20 / MXUC20c.2 MXUC40.2 / MXUC40c.2
VisiLine® IP	LXG	
VLG-22M.I / VLG-22C.I VLG-40M.I / VLG-40C.I	all models	
MXG		
MXGC20 / MXGC20c MXGC40 / MXGC40c		

Notice

For cameras with Camera Link® interface the feature is controlled via register.

Notice

The function *HDREnableTriggerAutoMode* is only supported by HX cameras.

Preparation

The following SDKs support the HDR feature:

Baumer GAPI SKD
Baumer GAPI SDK v2.1 (and higher)

Supported Programming Languages

The following programming languages enable HDR parameter adjustment:

Programming Languages
C++
C#

Table of Contents

1 General Information

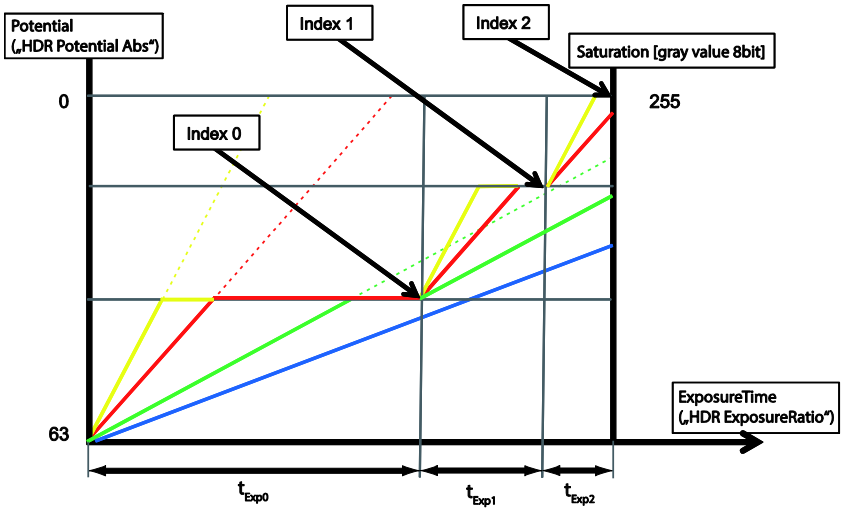
1.1 Normal HDR

The HDR mode limits the capacity of a pixel for a predetermined time. Capacity and time can be set for two knee points in the camera.

For each index a time (*HDRExposureRatio*) and a capacity (potential → *HDRPotentialAbs*) can be chosen.

The *HDRExposureRatio* value ranges from 1...255 (1%...100%), while the *HDRPotentialAbs* value –is within a range of 0...63 (0 = no limit; 63 = maximum limit).

The diagram below is for further explanation.



<u>Case 1 (low illumination) = blue line:</u>

The illumination which arrives at the sensor is very low, so no knee points are affected. There is no difference between HDR mode enabled / disabled.

<u>Case 2 (medium illumination) = green line:</u>

The illumination which arrives at the sensor is low, only first knee point is affected. If there was no HDR mode enabled, the small dotted green line would show the resulting gray value.

<u>Case 3 (high illumination) = red line:</u>

The illumination which arrives at the sensor is high, so that both knee points are affected. As the red dotted line shows, the pixel would be saturated after just half of the total exposure time, if no HDR mode was enabled.

<u>Case 4 (very high illumination) = yellow line:</u>

The illumination which arrives at the sensor is very high, so that both knee points are affected and the pixel is still saturated. If there was no HDR mode enabled, the pixel would be saturated after just one third of the total exposure time, as can be seen from the dotted yellow line.

A pixel of a camera –may be compared with a container to be filled with photons.

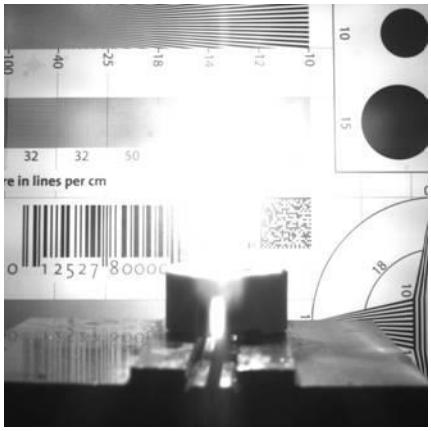
The “container” will take in photons until the limit (HDRPotentialABS Index 0) is reached and no more photons can be absorbed. After a certain time (HDRExposureRAtio Index 0), the container capacity is raised to HDRPotentialAb Index 1 to allow for the intake of more photons.

This is continued until the total exposure time is reached.

HDR mode allows darkening very bright (saturated) pixels, but it is not possible to make dark pixels brighter. Neither it is possible to adjust the values for *HDRPotentialAbs* and *HDRExposureRatio Index 2*. *HDRPotentialAbs* is always 0 while *HDRExposureRatio Index 2 = total exposure time - HDRExposureRatio Index 1 - HDRExposureRatio Index 0* is automatically calculated.

When applying HDR mode on an image, adjust the total exposure time to a suitable value until the dark areas of the image meet your requirements. Then the saturated pixels have to be "damped" which is done by changing the pixel limit or the exposure time of a certain time slot.

Example



HDR Off

HDR On

1.2 Trigger controlled HDR

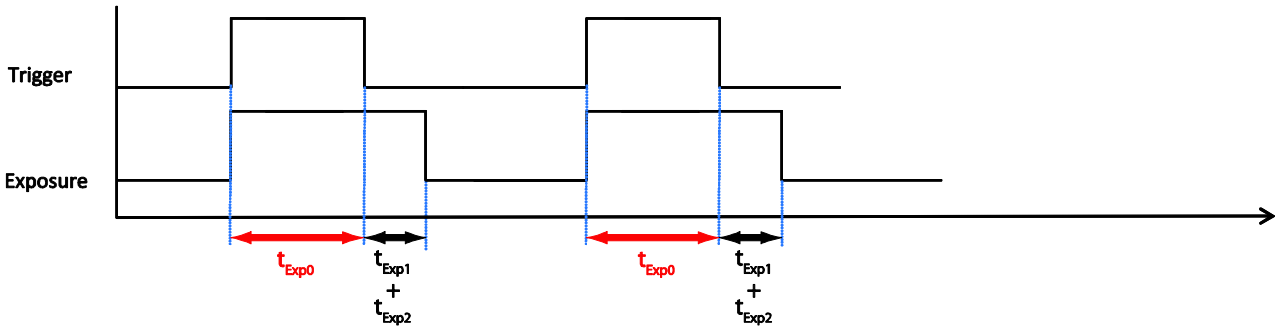
The *HDREnableTriggerAutoMode* is used where chosing ExposureMode *TriggerControlled* or *TriggerWidth*. This means, *HDREnableTriggerAutoMode* is only recommended where the ExposureTime is controlled by the pulse width of the trigger source.

In general, with *HDREnableTriggerAutoMode* being enabled, the ExposureTime for the first exposure slot (t_{Exp0}) is trigger controlled while the other two exposure time sections are automatically calculated according to HDR settings.

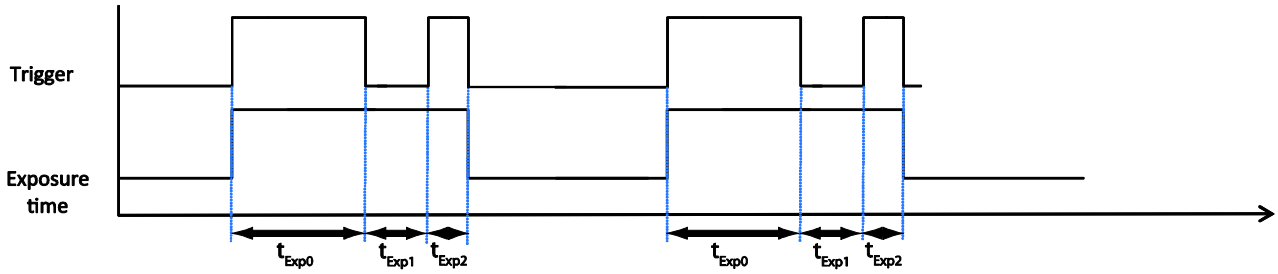
HDREnableTriggerAutoMode disabled entails trigger-controlled ExposureTime for all three sections.

The diagrams below show all four options the examples apply trigger activation by *RisingEdge*).

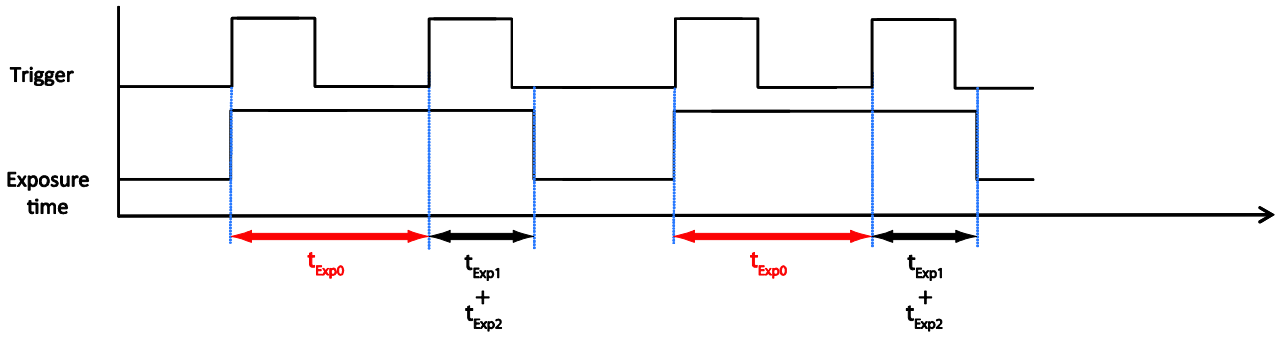
HDR EnableTriggerAutoMode = True / ExposureMode = TriggerWidth



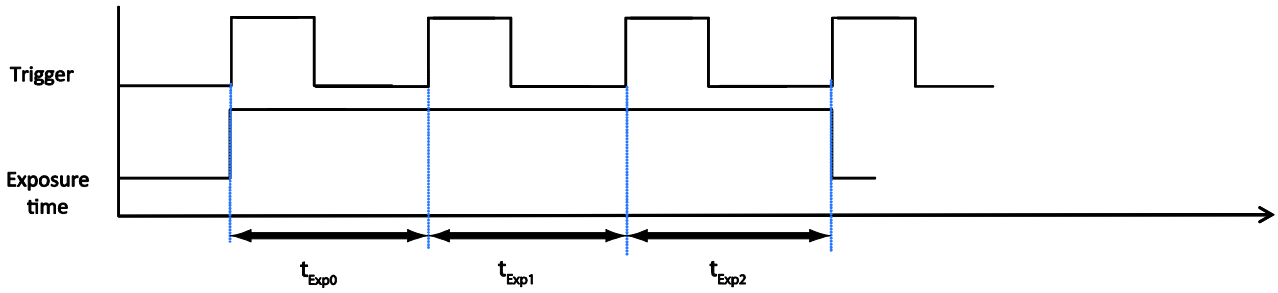
HDR EnableTriggerAutoMode = False / ExposureMode = TriggerWidth



HDR EnableTriggerAutoMode = True / ExposureMode = TriggerControlled



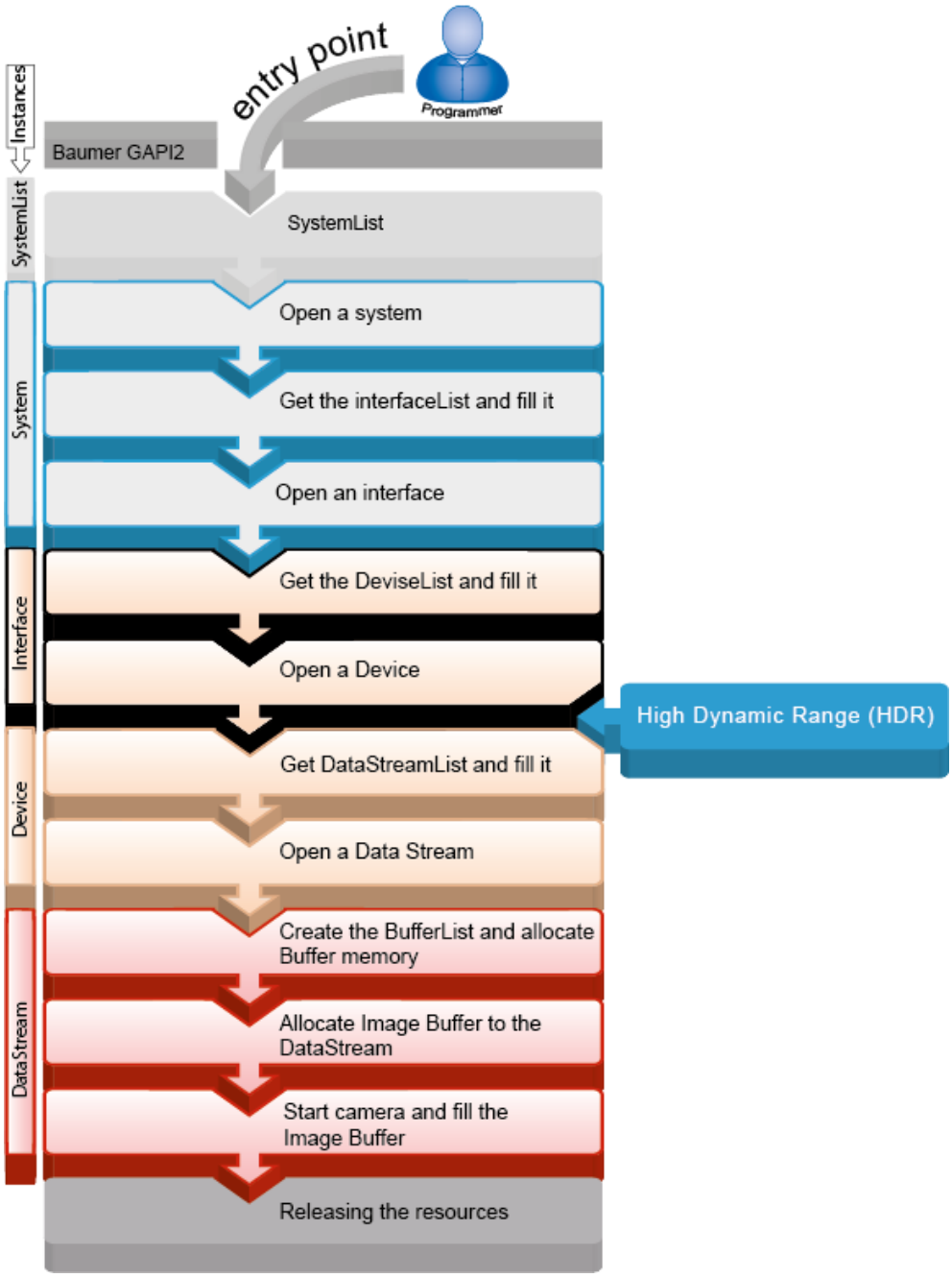
HDR EnableTriggerAutoMode = False / ExposureMode = TriggerControlled



2 HDR Settings

The illustration below shows the HDR feature location in the program sequence.

Notice
For better clarity, the code for initialization and image grabbing is removed. Only HDR parameter changes are shown.

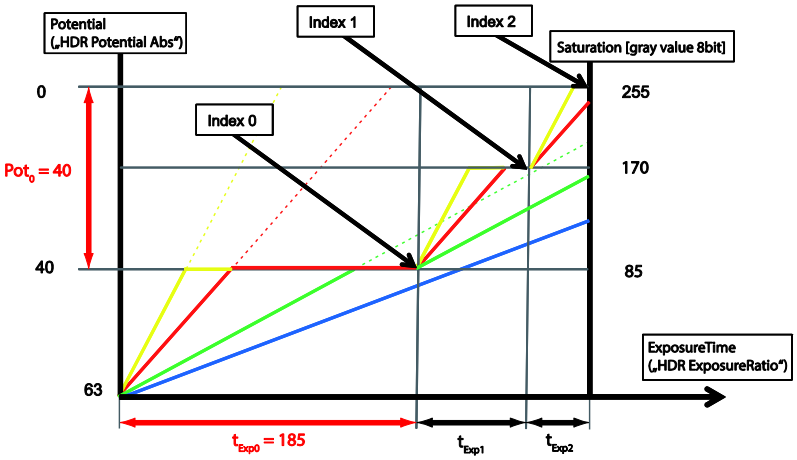


2.1 Normal HDR

2.1.1 C++

This chapter describes setting of *Normal HDR* in C++.

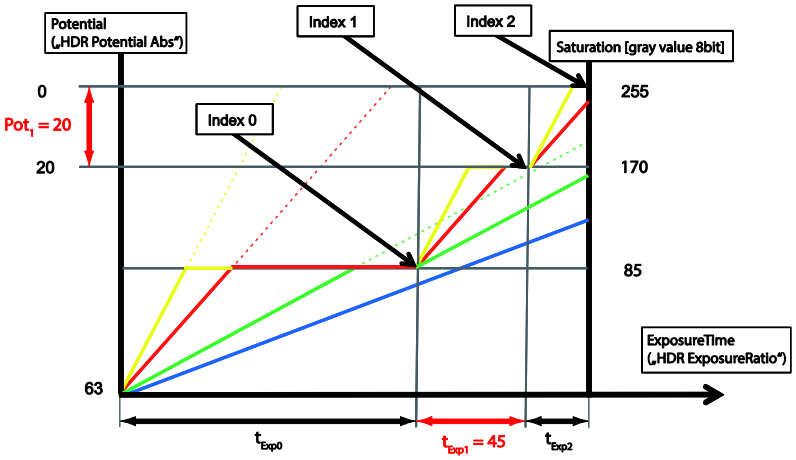
```
SystemList
Open a System
Get the InterfaceList and fill it
Open an Interface
Get the DeviceList and fill it
Open a Device

//Device Parameter Setup

pDevice->GetRemoteNode("TriggerMode")->SetString("Off");

pDevice->GetRemoteNode("ExposureTime")->SetDouble(10000.0);

std::cout << " ExposureTime : "
          << pDevice->GetRemoteNode("ExposureTime")->GetDouble()
          << std::endl


//HDR parameter change

std::cout << "HDR parameter change" << std::endl;

pDevice->GetRemoteNode("HDREnable")->SetBool(true);

std::cout << "HDREnable : "
          << pDevice->GetRemoteNode("HDREnable")->GetBool()
          << std::endl;


pDevice->GetRemoteNode("HDRIndex")->SetInt(0);

std::cout << "HDRIndex: "
          << pDevice->GetRemoteNode("HDRIndex")->GetInt()
          << std::endl;

pDevice->GetRemoteNode("HDRExposureRatio")->SetInt(185); //t_Exp_0

std::cout << "HDRExposureRatio: "
          << pDevice->GetRemoteNode("HDRExposureRatio")->GetInt()
          << std::endl;

std::cout << "HDRExposureRatioPercent : "
          << pDevice->GetRemoteNode("HDRExposureRatioPercent")->GetDouble()
          << std::endl;

pDevice->GetRemoteNode("HDRPotentialAbs")->SetInt(40); //Pot_0
std::cout << "HDRPotentialAbs : "
          << pDevice->GetRemoteNode("HDRPotentialAbs")->GetInt()
          << std::endl;
```




```
pDevice->GetRemoteNode("HDRIndex")->SetInt(1);
std::cout << "HDRIndex: "
          << pDevice->GetRemoteNode("HDRIndex")->GetInt()
          << std::endl;
pDevice->GetRemoteNode("HDRExposureRatio")->SetInt(45); //t_Exp_1
std::cout << "HDRExposureRatio: "
          << pDevice->GetRemoteNode("HDRExposureRatio")->GetInt()
          << std::endl;
std::cout << "HDRExposureRatioPercent: "
          << pDevice->GetRemoteNode("HDRExposureRatioPercent")->GetDouble()
          << std::endl;
pDevice->GetRemoteNode("HDRPotentialAbs")->SetInt(20); //Pot_1
std::cout << "HDRPotentialAbs: "
          << pDevice->GetRemoteNode("HDRPotentialAbs")->GetInt()
          << std::endl;
```

Notice

Value t_{Exp2} is automatically calculated in the camera.
($t_{Exp2} = t_{exposure} - t_{Exp0} - t_{Exp1}$
Pot_2 is always 0.



```
Get DataStreamList and fill it
Open a Data Stream
Create the BufferList and allocate Buffer memory
Allocate Image Buffer to the DataStream
Start Camera and fill the Image Buffer
Releasing the resources
```

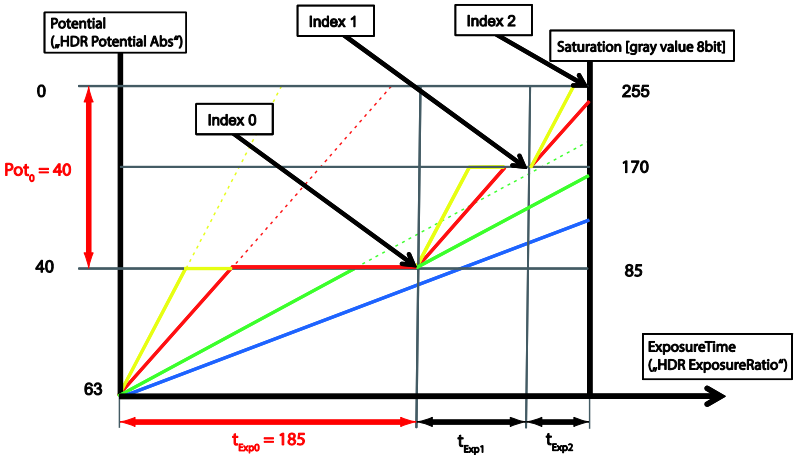
```
ExposureTime            : 10000

HDR parameter change
  HDREnable             : 1
  HDRIndex              : 0
  HDRExposureRatio      : 185
  HDRExposureRatio
  Percent               : 72.54
  HDRPotentialAbs       : 40

HDRIndex                : 1
  HDRExposureRatio      : 45
  HDRExposureRatio
  Percent               : 17.64
  HDRPotentialAbs       : 20
```

Console Output (C++)

2.1.2 C#

This chapter describes setting of *Normal HDR* in C#.

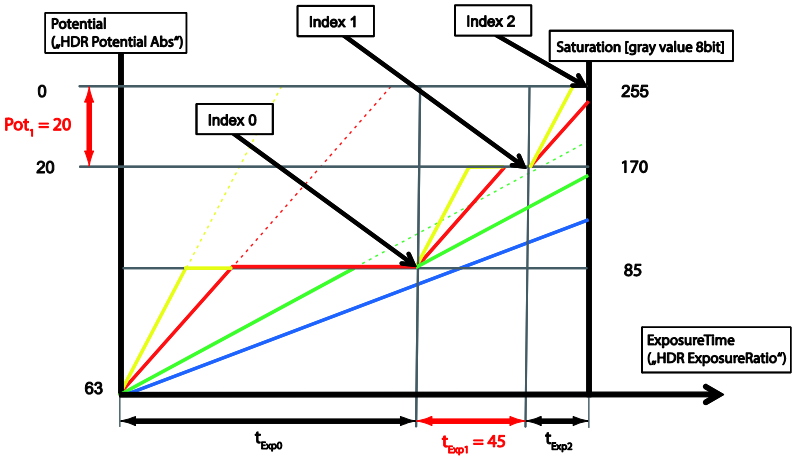
```
SystemList
Open a System
Get the InterfaceList and fill it
Open an Interface
Get the DeviceList and fill it
Open a Device


//Device Parameter Setup

mDevice.RemoteNodeList["TriggerMode"].Value = "Off";

mDevice.RemoteNodeList["ExposureTime"].Value = 10000.0;

System.Console.Write("ExposureTime : {0}\n",
   (double)mDevice.RemoteNodeList["ExposureTime"].Value);

System.Console.Write("\n");


//HDR parameter change

System.Console.Write("HDR parameter change\n");

mDevice.RemoteNodeList["HDREnable"].Value = true;

System.Console.Write("  HDREnable : {0}\n",
   (bool)mDevice.RemoteNodeList["HDREnable"].Value);


//only HXG

mDevice.RemoteNodeList["HDREnableTriggerAutoMode"].Value = false;

System.Console.Write("HDREnableTriggerAutoMode : {0}\n",
   (bool)mDevice.RemoteNodeList["HDREnableTriggerAutoMode"].Value);


mDevice.RemoteNodeList["HDRIndex"].Value = (long)0;

System.Console.Write("HDRIndex : {0}\n",
   (long)mDevice.RemoteNodeList["HDRIndex"].Value);

mDevice.RemoteNodeList["HDRExposureRatio"].Value = (long)185; //t_Exp_0

System.Console.Write("HDRExposureRatio : {0}\n",
   (long)mDevice.RemoteNodeList["HDRExposureRatio"].Value);

System.Console.Write("HDRExposureRatioPercent : {0}\n",
   (double)mDevice.RemoteNodeList["HDRExposureRatioPercent"].Value);

mDevice.RemoteNodeList["HDRExposureRatio"].Value = (long)40; //Pot_0
System.Console.Write("HDRPotentialAbs : {0}\n",
   (long)mDevice.RemoteNodeList["HDRPotentialAbs"].Value);
```




```
mDevice.RemoteNodeList["HDRIndex"].Value = (long)1;

System.Console.Write("HDRIndex : {0}\n",
   (long)mDeviceRemoteNodeList["HDRIndex"].Value);

mDevice.RemoteNodeList["HDRExposureRatio"].Value = (long)45; //t_Exp_1

System.Console.Write("HDRExposureRatio : {0}\n",
   (long)mDevice.RemoteNodeList["HDRExposureRatio"].Value);

System.Console.Write("HDRExposureRatioPercent : {0}\n",
   (double)mDevice.RemoteNodeList["HDRExposureRatioPercent"].Value);

mDevice.RemoteNodeList["HDRPotentialAbs"].Value = (long)20; //Pot_1

System.Console.Write("HDRPotentialAbs : {0}\n",
   (long)mDevice.RemoteNodeList["HDRPotentialAbs"].Value);
System.Console.Write("\n");
```

Notice

Value for t_{Exp2} is automatically calculated in the camera.
($t_{Exp2} = t_{exposure} - t_{Exp0} - t_{Exp1}$)
Pot_2 is always 0.



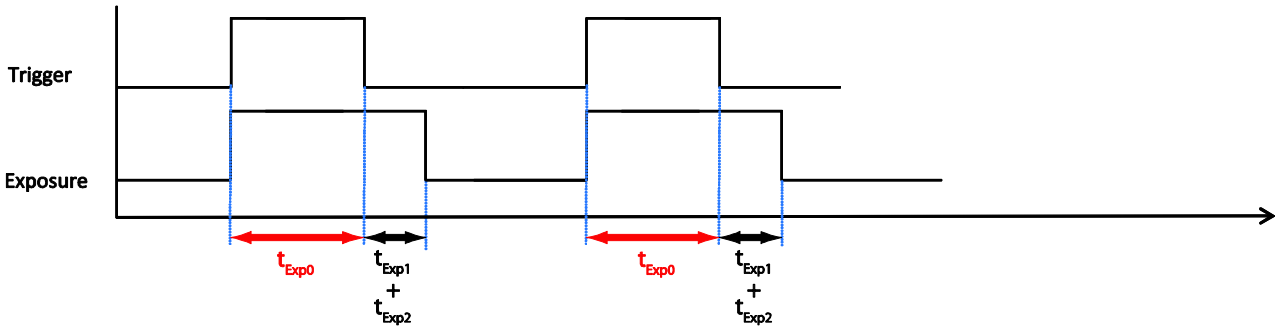
```
Get DataStreamList and fill it
Open a Data Stream
Create the BufferList and allocate Buffer memory
Allocate Image Buffer to the DataStream
Start Camera and fill the Image Buffer
Releasing the resources
```

```
ExposureTime            : 10000

HDR parameter change
  HDREnable             : True
  HDRIndex              : 0
  HDRExposureRatio      : 185
  HDRExposureRatio
  Percent               : 72.54
  HDRPotentialAbs       : 40

HDRIndex                : 1
  HDRExposureRatio      : 45
  HDRExposureRatio
  Percent               : 17.64
  HDRPotentialAbs       : 20
```

Console Output (C#)

2.2 Trigger controlled HDR

2.2.1 HDREnableTriggerAutoMode = True & ExposureMode = TriggerWidth



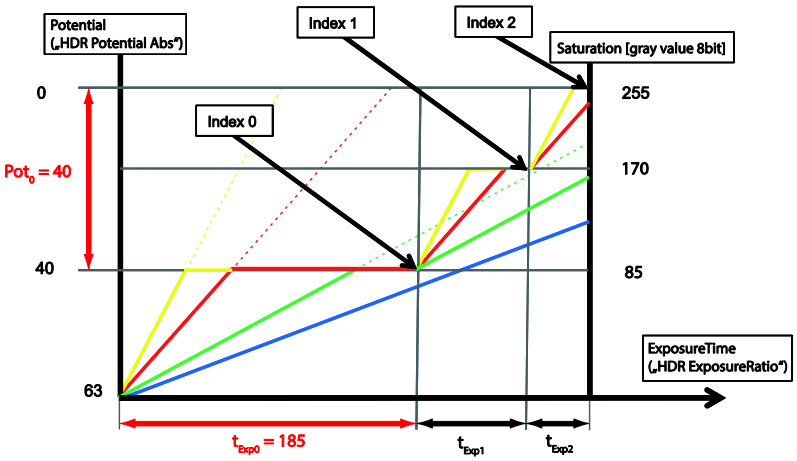
The exposure time for the first exposure slot (t_{Exp0}) is controlled while the other two exposure time sections (t_{Exp0}, t_{Exp1}) are calculated automatically according to HDR settings.

2.2.1.1 C++

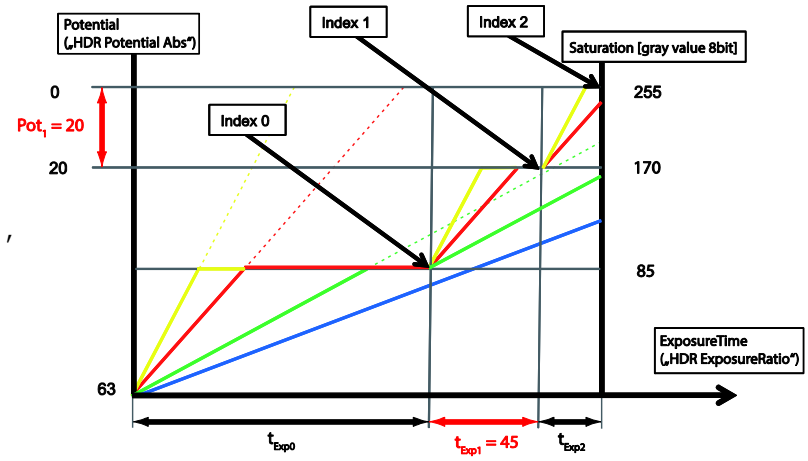
This chapter describes setting of *HDREnableTriggerAutoMode* = *True* & *ExposureMode* = *TriggerWidth* in C ++.

```
SystemList
Open a System
Get the InterfaceList and fill it
Open an Interface
Get the DeviceList and fill it
Open a Device


//Device Parameter Setup

if(pDevice->GetRemoteNode("ExposureMode")->GetValue() == "Timed")

{

    pDevice->GetRemoteNode("TriggerMode")->SetString("On");

}

std::cout << "TriggerMode: "
          << pDevice->GetRemoteNode("TriggerMode")->GetValue()
          << std::endl;


pDevice->GetRemoteNode("TriggerSource")->SetString("Line0");

std::cout << "TriggerSource: "
          << pDevice->GetRemoteNode("TriggerSource")->GetValue()
          << std::endl;
```

```
pDevice->GetRemoteNode("TriggerActivation")->SetString("RisingEdge");

std::cout << "TriggerActivation: "
          << pDevice->GetRemoteNode("TriggerActivation")->GetValue()
          << std::endl;


pDevice->GetRemoteNode("ExposureMode")->SetString("TriggerWidth");

std::cout << "ExposureMode:"
          << pDevice->GetRemoteNode("ExposureMode")->GetValue()
          << std::endl;


//HDR parameter change

std::cout << "HDR parameter change" << std::endl;

pDevice->GetRemoteNode("HDREnable")->SetBool(true);

std::cout << "HDREnable:"
          << pDevice->GetRemoteNode("HDREnable")->GetBool()
          << std::endl;


//only HXG

pDevice->GetRemoteNode("HDREnableTriggerAutoMode")->SetBool(true);

std::cout << "HDREnableTriggerAutoMode:"
          << pDevice->GetRemoteNode("HDREnableTriggerAutoMode")->GetBool()
          << std::endl;


pDevice->GetRemoteNode("HDRIndex")->SetInt(0);

std::cout << "HDRIndex:"
          << pDevice->GetRemoteNode("HDRIndex")->GetInt()
          << std::endl;

pDevice->GetRemoteNode("HDRExposureRatio")->SetInt(185); //t_Exp_0

std::cout << "HDRExposureRatio: "
          << pDevice->GetRemoteNode("HDRExposureRatio")->GetInt()
          << std::endl;

std::cout << "HDRExposureRatioPercent: "
          << pDevice->GetRemoteNode("HDRExposureRatioPercent")->GetDouble()
          << std::endl;

pDevice->GetRemoteNode("HDRPotentialAbs")->SetInt(40); //Pot_0

std::cout << "HDRPotentialAbs:"
          << pDevice->GetRemoteNode("HDRPotentialAbs")->GetInt()
          << std::endl;
```




```
pDevice->GetRemoteNode("HDRIndex")->SetInt(1);

std::cout << "HDRIndex:"
          << pDevice->GetRemoteNode("HDRIndex")->GetInt()
          << std::endl;

pDevice->GetRemoteNode("HDRExposureRatio")->SetInt(45); //t_Exp_1

std::cout << "HDRExposureRatio:"
          << pDevice->GetRemoteNode("HDRExposureRatio")->GetInt()
          << std::endl;

std::cout << "HDRExposureRatioPercent:"
          << pDevice->GetRemoteNode("HDRExposureRatioPercent")->GetDouble()
          << std::endl;

pDevice->GetRemoteNode("HDRPotentialAbs")->SetInt(20); //Pot_1

std::cout << "HDRPotentialAbs:"
          << pDevice->GetRemoteNode("HDRPotentialAbs")->GetInt()
          << std::endl;
```



```
Get DataStreamList and fill it
Open a Data Stream
Create the BufferList and allocate Buffer memory
Allocate Image Buffer to the DataStream
Start Camera and fill the Image Buffer
Releasing the resources
```

```
TriggerMode: On

TriggerSource: Line0

TriggerActivation: RisingEdge

ExposureMode: TriggerWidth

HDR parameter change
HDREnable: True
HDREnableTriggerAutoMode: True
HDRIndex: 0
HDRExposureRatio: 185
HDRExposureRatioPercent: 72.83
HDRPotentialAbs: 40

HDRIndex: 1
HDRExposureRatio: 45
HDRExposureRatioPercent: 17.72
HDRPotentialAbs: 20
```

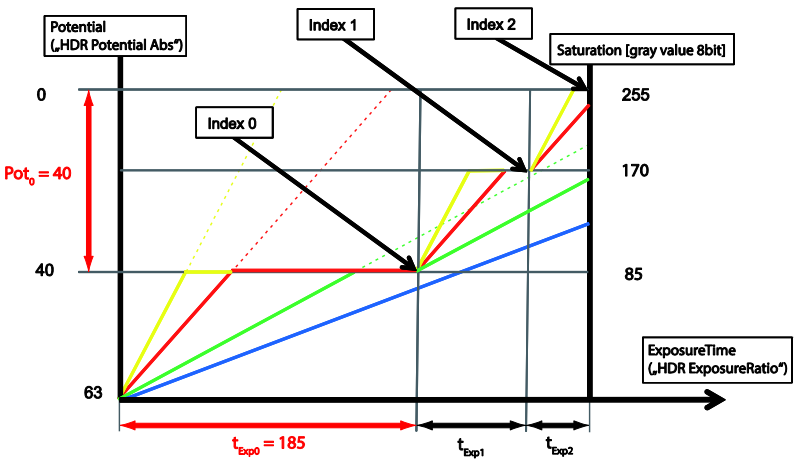
Console Output (C++)

2.2.1.2 C#

This chapter describes setting of *HDREnableTriggerAutoMode = True & ExposureMode = TriggerWidth* in C#.

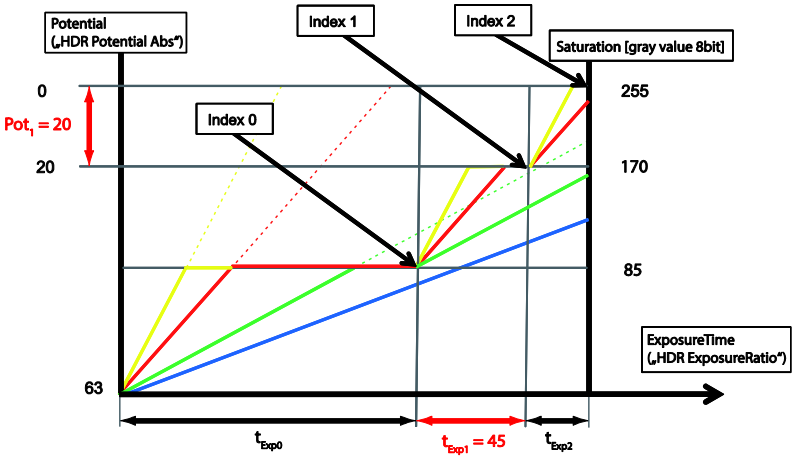
```
SystemList
Open a System
Get the InterfaceList and fill it
Open an Interface
Get the DeviceList and fill it
Open a Device


//Device Parameter Setup

if ((string)mDevice.RemoteNodeList["ExposureMode"].Value == "Timed")

{

    mDevice.RemoteNodeList["TriggerMode"].Value = "On";

}

System.Console.Write("TriggerMode: {0}\n\n",
   (string)mDevice.RemoteNodeList["TriggerMode"].Value);


mDevice.RemoteNodeList["TriggerSource"].Value = "Line0";

System.Console.Write("TriggerSource : {0}\n\n",
   (string)mDevice.RemoteNodeList["TriggerSource"].Value);


mDevice.RemoteNodeList["TriggerActivation"].Value = "RisingEdge";

System.Console.Write("  TriggerActivation: {0}\n\n",
   (string)mDevice.RemoteNodeList["TriggerActivation"].Value);


mDevice.RemoteNodeList["ExposureMode"].Value = "TriggerWidth";

System.Console.Write("ExposureMode : {0}\n\n",
   (string)mDevice.RemoteNodeList["ExposureMode"].Value);


//HDR parameter change

System.Console.Write("HDR parameter change\n");

mDevice.RemoteNodeList["HDREnable"].Value = true;

System.Console.Write(" HDREnable : {0}\n",
   (bool)mDevice.RemoteNodeList["HDREnable"].Value);
```

```
//only HXG

mDevice.RemoteNodeList["HDREnableTriggerAutoMode"].Value = true;

System.Console.Write("  HDREnableTriggerAutoMode  : {0}\n",
   (bool)mDevice.RemoteNodeList["HDREnableTriggerAutoMode"].Value);


mDevice.RemoteNodeList["HDRIndex"].Value = (long)0;

System.Console.Write("  HDRIndex : {0}\n",
   (long)mDevice.RemoteNodeList["HDRIndex"].Value);


mDevice.RemoteNodeList["HDRExposureRatio"].Value = (long)185; //t_Exp_0

System.Console.Write(" HDRExposureRatio: {0}\n",
   (long)mDevice.RemoteNodeList["HDRExposureRatio"].Value);

System.Console.Write("HDRExposureRatioPercent: {0:F2}\n",
   (double)mDevice.RemoteNodeList["HDRExposureRatioPercent"].Value);

mDevice.RemoteNodeList["HDRPotentialAbs"].Value = (long)40; //Pot_0

System.Console.Write("HDRPotentialAbs : {0}\n",
   (long)mDevice.RemoteNodeList["HDRPotentialAbs"].Value);
```




```
mDevice.RemoteNodeList["HDRIndex"].Value = (long)1;

System.Console.Write("HDRIndex: {0}\n",
   (long)mDevice.RemoteNodeList["HDRIndex"].Value);

mDevice.RemoteNodeList["HDRExposureRatio"].Value = (long)45; //Pot_1

System.Console.Write("HDRExposureRatio : {0}\n",
   (long)mDevice.RemoteNodeList["HDRExposureRatio"].Value);

System.Console.Write("HDRExposureRatioPercent: {0:F2}\n",
   (double)mDevice.RemoteNodeList["HDRExposureRatioPercent"].Value);

mDevice.RemoteNodeList["HDRPotentialAbs"].Value = (long)20; //t_Exp_1
```

```
System.Console.Write("HDRPotentialAbs: {0}\n",
   (long)mDevice.RemoteNodeList["HDRPotentialAbs"].Value);

System.Console.Write("\n");
```

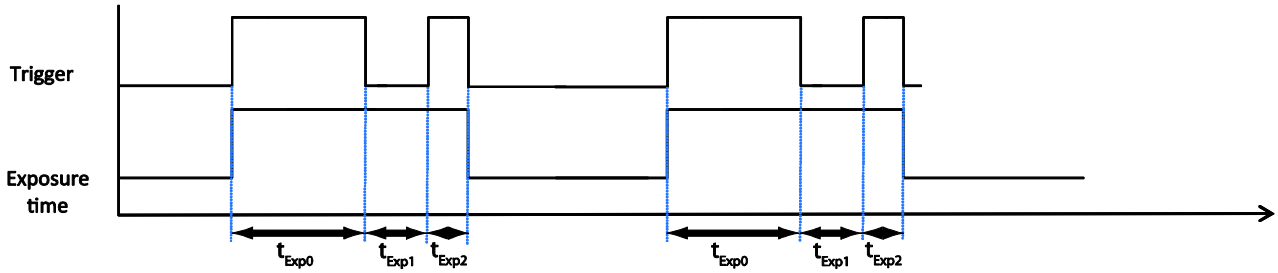



```
Get DataStreamList and fill it
Open a Data Stream
Create the BufferList and allocate Buffer memory
Allocate Image Buffer to the DataStream
Start Camera and fill the Image Buffer
Releasing the resources
```

```
TriggerMode: On

TriggerSource: Line0
TriggerActivation: RisingEdge
ExposureMode: TriggerWidth
HDR parameter change
HDREnable: True
HDREnableTriggerAutoMode:
True

HDRIndex: 0
HDRExposureRatio: 185
HDRExposureRatioPercent:
72.83
HDRPotentialAbs: 40
HDRIndex: 1
HDRExposureRatio: 45
HDRExposureRatioPercent:
17,72
HDRPotentialAbs: 20
```

Console Output (C#)

2.2.2 EnableTriggerAutoMode = False & ExposureMode = TriggerWidth



The exposure times for all three exposure slots are trigger controlled.

2.2.2.1 C++

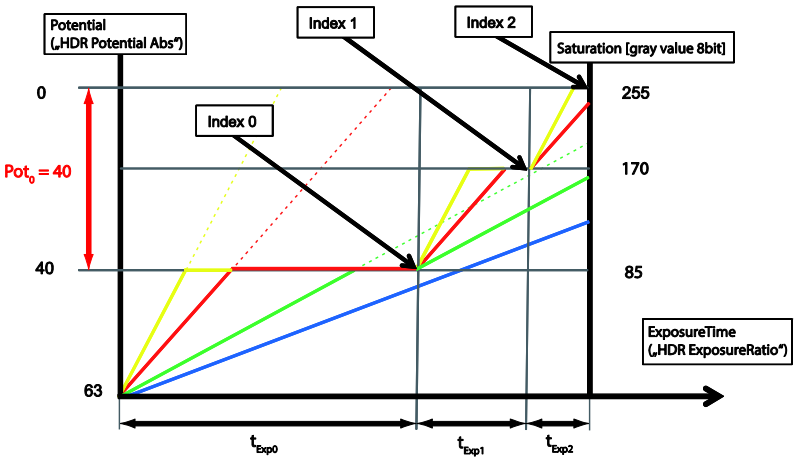
This chapter describes setting of *EnableTriggerAutoMode* = *False* & *ExposureMode* = *TriggerWidth* in C++.

```
SystemList
Open a System
Get the InterfaceList and fill it
Open an Interface
Get the DeviceList and fill it
Open a Device


//Device Parameter Setup

if(pDevice->GetRemoteNode("ExposureMode")->GetValue() == "Timed")

{

    pDevice->GetRemoteNode("TriggerMode")->SetString("On");

}

std::cout << "TriggerMode: "
          << pDevice->GetRemoteNode("TriggerMode")->GetValue()
          << std::endl;


pDevice->GetRemoteNode("TriggerSource")->SetString("Line0");

std::cout << "TriggerSource: "
          << pDevice->GetRemoteNode("TriggerSource")->GetValue()
          << std::endl;


pDevice->GetRemoteNode("TriggerActivation")->SetString("RisingEdge");

std::cout << "  TriggerActivation: "
          << pDevice->GetRemoteNode("TriggerActivation")->GetValue()
          << std::endl;
```

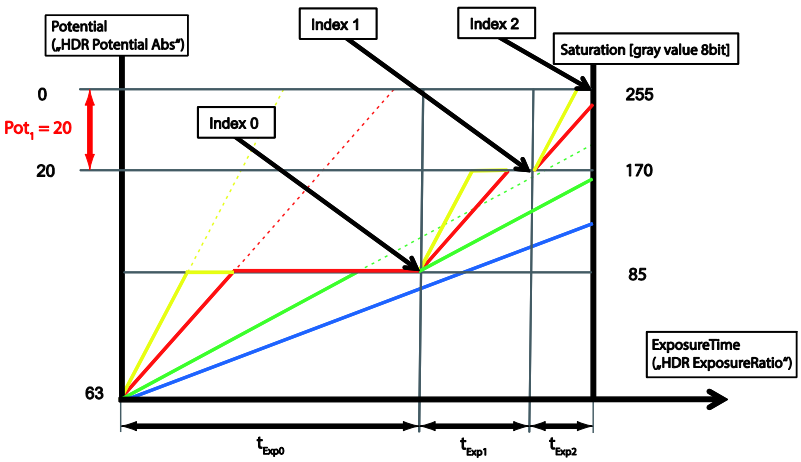
```
pDevice->GetRemoteNode("ExposureMode")->SetString("TriggerWidth");

std::cout << "ExposureMode: "
          << pDevice->GetRemoteNode("ExposureMode")->GetValue()
          << std::endl;


//HDR parameter change

std::cout << "HDR parameter change" << std::endl;

pDevice->GetRemoteNode("HDREnable")->SetBool(true);

std::cout << "HDREnable : "
          << pDevice->GetRemoteNode("HDREnable")->GetBool()
          << std::endl;


//only HXG

pDevice->GetRemoteNode("HDREnableTriggerAutoMode")->SetBool(false);

std::cout << "HDREnableTriggerAutoMode: "
          << pDevice->GetRemoteNode("HDREnableTriggerAutoMode")->GetBool()
          << std::endl;


pDevice->GetRemoteNode("HDRIndex")->SetInt(0);

std::cout << "HDRIndex: "
          << pDevice->GetRemoteNode("HDRIndex")->GetInt()
          << std::endl;


//pDevice->GetRemoteNode("HDRExposureRatio")->SetInt(185); //t_expo_0

//std::cout << "HDRExposureRatio: "
            << pDevice->GetRemoteNode("HDRExposureRatio")->GetInt()
            << std::endl;

//std::cout << "HDRExposureRatioPercent :"
            << pDevice->GetRemoteNode("HDRExposureRatioPercent")->GetDouble()
            << std::endl;

pDevice->GetRemoteNode("HDRPotentialAbs")->SetInt(40); //Pot_0

std::cout << "HDRPotentialAbs: "
          << pDevice->GetRemoteNode("HDRPotentialAbs")->GetInt()
          << std::endl;
```

Notice

If `HDREnableTriggerAutoMode = False, HDRExposureRatio` and `HDRExposureRatioPercent` will not be effected.




```
pDevice->GetRemoteNode("HDRIndex")->SetInt(1);

std::cout << "HDRIndex: "
          << pDevice->GetRemoteNode("HDRIndex")->GetInt()
          << std::endl;


//pDevice->GetRemoteNode("HDRExposureRatio")->SetInt(45); //t_expo_1

//std::cout << "HDRExposureRatio: "
            << pDevice->GetRemoteNode("HDRExposureRatio")->GetInt()
            << std::endl;

//std::cout << "HDRExposureRatioPercent: "
            << pDevice->GetRemoteNode("HDRExposureRatioPercent")->GetDouble()
            << std::endl;

pDevice->GetRemoteNode("HDRPotentialAbs")->SetInt(20); //Pot_1

std::cout << "HDRPotentialAbs: "
          << pDevice->GetRemoteNode("HDRPotentialAbs")->GetInt()
          << std::endl;
```



```
Get DataStreamList and fill it
Open a Data Stream
Create the BufferList and allocate Buffer memory
Allocate Image Buffer to the DataStream
Start Camera and fill the Image Buffer
Releasing the resources
```

```
TriggerMode: On

TriggerSource: Line0
TriggerActivation: RisingEdge
ExposureMode: TriggerWidth
HDR parameter change
HDREnable: True

HDREnableTriggerAutoMode:
False
HDRIndex: 0
HDRPotentialAbs: 40
HDRIndex: 1
HDRPotentialAbs: 20
```

Console Output (C++)

2.2.2.2 C#

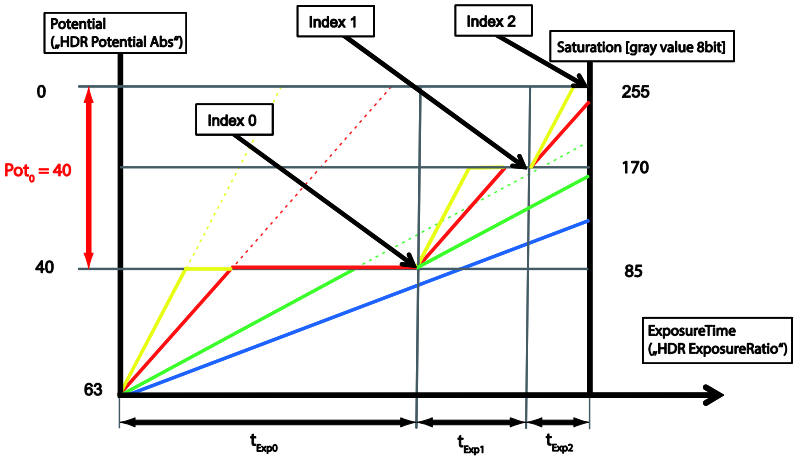
This chapter describes setting of *EnableTriggerAutoMode* = *False* & *ExposureMode* = *TriggerWidth* in C#.

```
SystemList
Open a System
Get the InterfaceList and fill it
Open an Interface
Get the DeviceList and fill it
Open a Device


//Device Parameter Setup

if ((string)mDevice.RemoteNodeList["ExposureMode"].Value == "Timed")

{

    mDevice.RemoteNodeList["TriggerMode"].Value = "On";

}

System.Console.Write("TriggerMode: {0}\n\n",
   (string)mDevice.RemoteNodeList["TriggerMode"].Value);


mDevice.RemoteNodeList["TriggerSource"].Value = "Line0";

System.Console.Write("TriggerSource: {0}\n\n",
   (string)mDevice.RemoteNodeList["TriggerSource"].Value);


mDevice.RemoteNodeList["TriggerActivation"].Value = "RisingEdge";

System.Console.Write("TriggerActivation: {0}\n\n",
   (string)mDevice.RemoteNodeList["TriggerActivation"].Value);


mDevice.RemoteNodeList["ExposureMode"].Value = "TriggerWidth";

System.Console.Write("ExposureMode: {0}\n\n",
   (string)mDevice.RemoteNodeList["ExposureMode"].Value);


//HDR parameter change

System.Console.Write("HDR parameter change\n");

mDevice.RemoteNodeList["HDREnable"].Value = true;

System.Console.Write("HDREnable: {0}\n",
   (bool)mDevice.RemoteNodeList["HDREnable"].Value);


//only HXG

mDevice.RemoteNodeList["HDREnableTriggerAutoMode"].Value = false;
```

```
System.Console.Write("HDREnableTriggerAutoMode : {0}\n",
   (bool)mDevice.RemoteNodeList["HDREnableTriggerAutoMode"].Value);


mDevice.RemoteNodeList["HDRIndex"].Value = (long)0;

System.Console.Write("HDRIndex: {0}\n",
   (long)mDevice.RemoteNodeList["HDRIndex"].Value);


//mDevice.RemoteNodeList["HDRExposureRatio"].Value = (long)185;

//System.Console.Write("HDRExposureRatio: {0}\n",
    (long)mDevice.RemoteNodeList["HDRExposureRatio"].Value);

//System.Console.Write("HDRExposureRatioPercent: {0:F2}\n",
    (double)mDevice.RemoteNodeList["HDRExposureRatioPercent"].Value);


mDevice.RemoteNodeList["HDRPotentialAbs"].Value = (long)40; //Pot_0

System.Console.Write("HDRPotentialAbs: {0}\n",
   (long)mDevice.RemoteNodeList["HDRPotentialAbs"].Value);
```

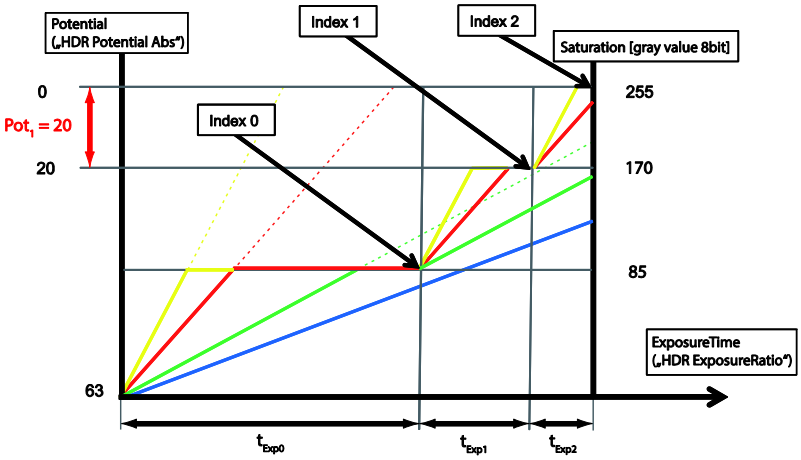
Notice

If `HDREnableTriggerAutoMode = False`, `HDRExposureRatio` and `HDRExposureRatioPercent` will not be effected.




```
mDevice.RemoteNodeList["HDRIndex"].Value = (long)1;

System.Console.Write("HDRIndex: {0}\n",
   (long)mDevice.RemoteNodeList["HDRIndex"].Value);

//mDevice.RemoteNodeList["HDRExposureRatio"].Value = (long)45;
```

```
//System.Console.Write("HDRExposureRatio: {0}\n",
    (long)mDevice.RemoteNodeList["HDRExposureRatio"].Value);

//System.Console.Write("HDRExposureRatioPercent: {0:F2}\n",
    (double)mDevice.RemoteNodeList["HDRExposureRatioPercent"].Value);


mDevice.RemoteNodeList["HDRPotentialAbs"].Value = (long)20; //Pot_1

System.Console.Write("HDRPotentialAbs: {0}\n",
   (long)mDevice.RemoteNodeList["HDRPotentialAbs"].Value);

System.Console.Write("\n");
```

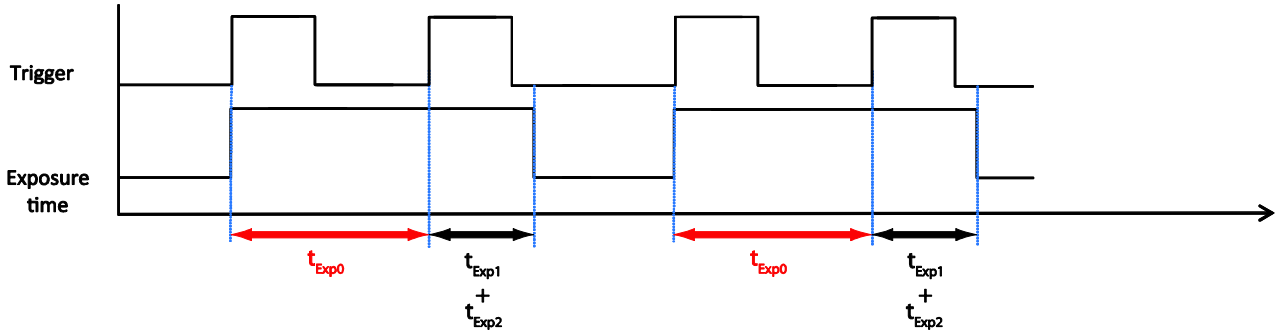



```
Get DataStreamList and fill it
Open a Data Stream
Create the BufferList and allocate Buffer memory
Allocate Image Buffer to the DataStream
Start Camera and fill the Image Buffer
Releasing the resources
```

```
TriggerMode: On

TriggerSource: Line0

TriggerActivation: RisingEdge

ExposureMode: TriggerWidth

HDR parameter change
HDREnable: True
HDREnableTriggerAutoMode: False
HDRIndex: 0
HDRPotentialAbs: 40
HDRIndex: 1
HDRPotentialAbs: 20
```

Console Output (C#)

2.2.3 HDREnableTriggerAutoMode = True & ExposureMode = TriggerControlled



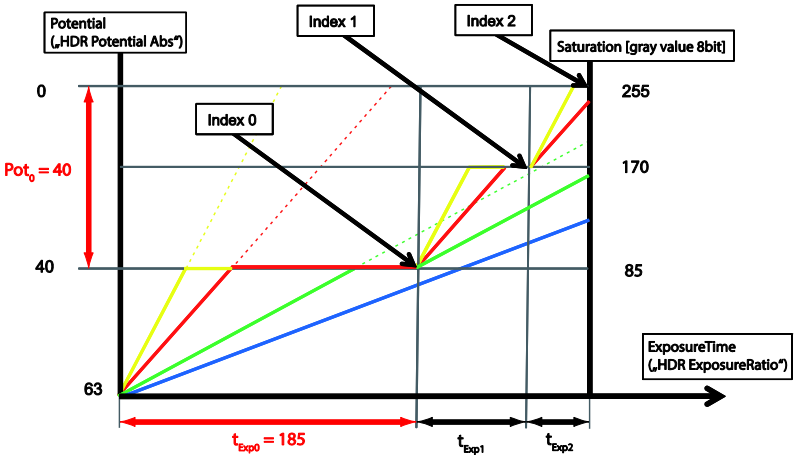
The ExposureTime for the first exposure slot (t_{Exp0}) is trigger- controlled while the other two exposure time sections are calculated automatically according to HDR settings.

2.2.3.1 C++

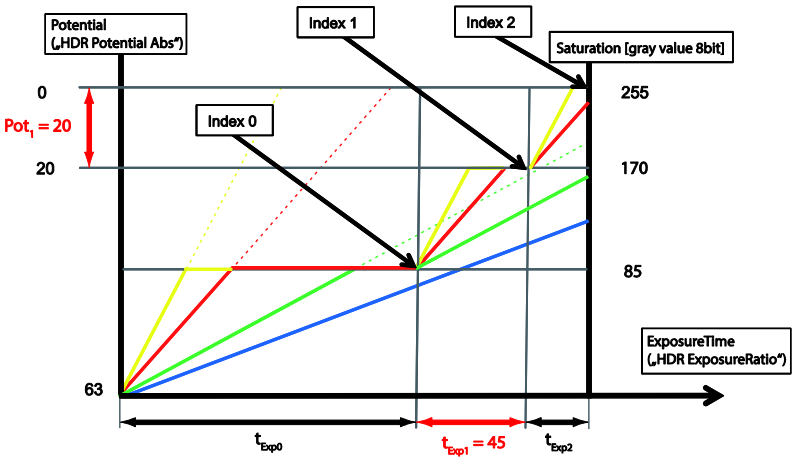
This chapter describes setting of *HDREnableTriggerAutoMode = True & ExposureMode = TriggerControlled* in C++.

```
SystemList
Open a System
Get the InterfaceList and fill it
Open an Interface
Get the DeviceList and fill it
Open a Device


//Device Parameter Setup

if(pDevice->GetRemoteNode("ExposureMode")->GetValue() == "Timed")

{

    pDevice->GetRemoteNode("TriggerMode")->SetString("On");

}

std::cout << "TriggerMode: "
          << pDevice->GetRemoteNode("TriggerMode")->GetValue()
          << std::endl;


pDevice->GetRemoteNode("TriggerSource")->SetString("Line0");

std::cout << "TriggerSource: "
          << pDevice->GetRemoteNode("TriggerSource")->GetValue()
          << std::endl;


pDevice->GetRemoteNode("TriggerActivation")->SetString("RisingEdge");
```

```
std::cout << "TriggerActivation: "
          << pDevice->GetRemoteNode("TriggerActivation")->GetValue()
          << std::endl;

pDevice->GetRemoteNode("ExposureMode")->SetString("TriggerControlled");

std::cout << "ExposureMode: "
          << pDevice->GetRemoteNode("ExposureMode")->GetValue()
          << std::endl;


//HDR parameter change

std::cout << "HDR parameter change" << std::endl;

pDevice->GetRemoteNode("HDREnable")->SetBool(true);

std::cout << "HDREnable: "
          << pDevice->GetRemoteNode("HDREnable")->GetBool()
          << std::endl;


//only HXG

pDevice->GetRemoteNode("HDREnableTriggerAutoMode")->SetBool(true);

std::cout << "HDREnableTriggerAutoMode: "
          << pDevice->GetRemoteNode("HDREnableTriggerAutoMode")->GetBool()
          << std::endl;


pDevice->GetRemoteNode("HDRIndex")->SetInt(0);

std::cout << "HDRIndex: "
          << pDevice->GetRemoteNode("HDRIndex")->GetInt()
          << std::endl;

pDevice->GetRemoteNode("HDRExposureRatio")->SetInt(185); //t_Exp_0

std::cout << "HDRExposureRatio: "
          << pDevice->GetRemoteNode("HDRExposureRatio")->GetInt()
          << std::endl;

std::cout << "HDRExposureRatioPercent: "
          << pDevice->GetRemoteNode("HDRExposureRatioPercent")->GetDouble()
          << std::endl;

pDevice->GetRemoteNode("HDRPotentialAbs")->SetInt(40); //Pot_0

std::cout << "HDRPotentialAbs :"
          << pDevice->GetRemoteNode("HDRPotentialAbs")->GetInt()
          << std::endl;
```




```
pDevice->GetRemoteNode("HDRIndex")->SetInt(1);

std::cout << "HDRIndex: "
          << pDevice->GetRemoteNode("HDRIndex")->GetInt()
          << std::endl;

pDevice->GetRemoteNode("HDRExposureRatio")->SetInt(45); //t_expo_1

std::cout << "HDRExposureRatio: "
          << pDevice->GetRemoteNode("HDRExposureRatio")->GetInt()
          << std::endl;

std::cout << "HDRExposureRatioPercent: "
          << pDevice->GetRemoteNode("HDRExposureRatioPercent")->GetDouble()
          << std::endl;

pDevice->GetRemoteNode("HDRPotentialAbs")->SetInt(20); //Pot_1

std::cout << "HDRPotentialAbs: "
          << pDevice->GetRemoteNode("HDRPotentialAbs")->GetInt()
          << std::endl;
```



```
Get DataStreamList and fill it
Open a Data Stream
Create the BufferList an allocate Buffer memory
Allocate Image Buffer to the DataStream
Start Camera and fill the Image Buffer
Releasing the resources
```

```
TriggerMode: On

TriggerSource: Line0

TriggerActivation: RisingEdge

ExposureMode: TriggerControlled

HDR parameter change
HDREnable: True
HDREnableTriggerAutoMode: True
HDRIndex: 0
HDRExposureRatio: 185
HDRExposureRatioPercent: 72.83
HDRPotentialAbs: 40
HDRIndex: 1
HDRExposureRatio: 45
HDRExposureRatioPercent: 17.72
HDRPotentialAbs: 20
```

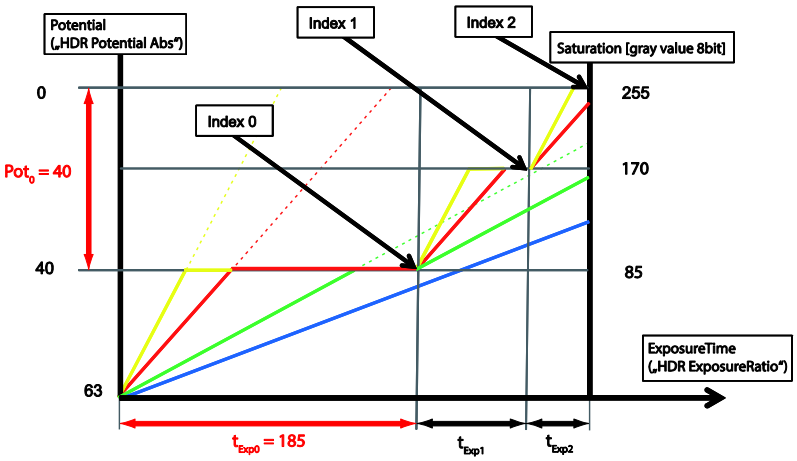
Console Output (C++)

2.2.3.2 C#

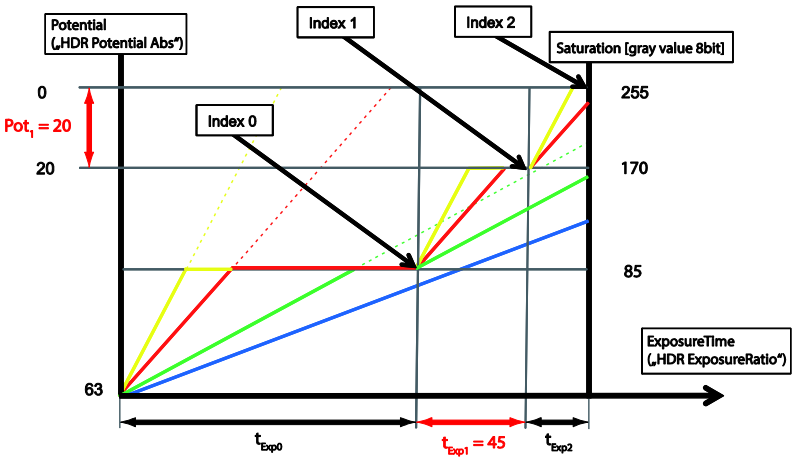
This chapter describes setting of *HDREnableTriggerAutoMode = True & ExposureMode = TriggerControlled* in C#.

```
SystemList
Open a System
Get the InterfaceList and fill it
Open an Interface
Get the DeviceList and fill it
Open a Device


//Device Parameter Setup

if ((string)mDevice.RemoteNodeList["ExposureMode"].Value == "Timed")

{

    mDevice.RemoteNodeList["TriggerMode"].Value = "On";

}

System.Console.Write("TriggerMode: {0}\n\n",
   (string)mDevice.RemoteNodeList["TriggerMode"].Value);


mDevice.RemoteNodeList["TriggerSource"].Value = "Line0";

System.Console.Write("TriggerSource: {0}\n\n",
   (string)mDevice.RemoteNodeList["TriggerSource"].Value);


mDevice.RemoteNodeList["TriggerActivation"].Value = "RisingEdge";

System.Console.Write("TriggerActivation: {0}\n\n",
   (string)mDevice.RemoteNodeList["TriggerActivation"].Value);


mDevice.RemoteNodeList["ExposureMode"].Value = "TriggerControlled";

System.Console.Write("ExposureMode: {0}\n\n",
   (string)mDevice.RemoteNodeList["ExposureMode"].Value);


//HDR parameter change

System.Console.Write("HDR parameter change\n");

mDevice.RemoteNodeList["HDREnable"].Value = true;

System.Console.Write("HDREnable: {0}\n",
   (bool)mDevice.RemoteNodeList["HDREnable"].Value);
```

```
//only HXG

mDevice.RemoteNodeList["HDREnableTriggerAutoMode"].Value = true;

System.Console.Write("  HDREnableTriggerAutoMode  : {0}\n",
   (bool)mDevice.RemoteNodeList["HDREnableTriggerAutoMode"].Value);

mDevice.RemoteNodeList["HDRIndex"].Value = (long)0;

System.Console.Write("HDRIndex: {0}\n",
   (long)mDevice.RemoteNodeList["HDRIndex"].Value);

mDevice.RemoteNodeList["HDRExposureRatio"].Value = (long)185; //t_Exp_0

System.Console.Write("HDRExposureRatio: {0}\n",
   (long)mDevice.RemoteNodeList["HDRExposureRatio"].Value);

System.Console.Write("HDRExposureRatioPercent: {0:F2}\n",
   (double)mDevice.RemoteNodeList["HDRExposureRatioPercent"].Value);

mDevice.RemoteNodeList["HDRPotentialAbs"].Value = (long)40; //Pot_0

System.Console.Write("HDRPotentialAbs: {0}\n",
   (long)mDevice.RemoteNodeList["HDRPotentialAbs"].Value);
```




```
mDevice.RemoteNodeList["HDRIndex"].Value = (long)1;

System.Console.Write("HDRIndex: {0}\n",
   (long)mDevice.RemoteNodeList["HDRIndex"].Value);

mDevice.RemoteNodeList["HDRExposureRatio"].Value = (long)45; //t_Exp_1

System.Console.Write("HDRExposureRatio: {0}\n",
   (long)mDevice.RemoteNodeList["HDRExposureRatio"].Value);

System.Console.Write("HDRExposureRatioPercent: {0:F2}\n",
   (double)mDevice.RemoteNodeList["HDRExposureRatioPercent"].Value);

mDevice.RemoteNodeList["HDRPotentialAbs"].Value = (long)20; //Pot_1

System.Console.Write("HDRPotentialAbs: {0}\n",
   (long)mDevice.RemoteNodeList["HDRPotentialAbs"].Value);

System.Console.Write("\n");
```

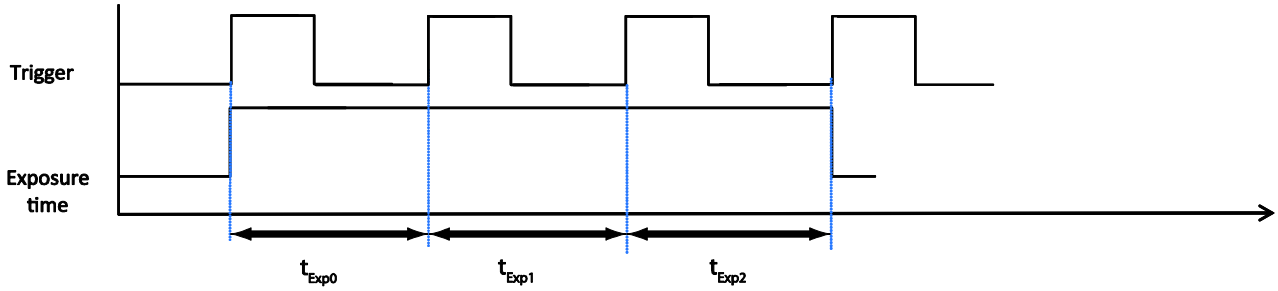



```
Get DataStreamList and fill it
Open a Data Stream
Create the BufferList an allocate Buffer memory
Allocate Image Buffer to the DataStream
Start Camera and fill the Image Buffer
Releasing the resources
```

```
TriggerMode: On

TriggerSource: Line0

TriggerActivation: RisingEdge

ExposureMode: TriggerControlled
HDR parameter change
HDREnable: True
HDREnableTriggerAutoMode: True
HDRIndex: 0
HDRExposureRatio: 185
HDRExposureRatioPercent: 72.83
HDRPotentialAbs: 40

HDRIndex: 1
HDRExposureRatio: 45
HDRExposureRatioPercent: 17.72
HDRPotentialAbs: 20
```

Console Output (C#)

2.2.4 HDREnableTriggerAutoMode = False & ExposureMode = TriggerControlled



The exposure times for all three exposure slots (t_{Exp0}, t_{Exp1}, t_{Exp2}) are trigger controlled.

2.2.4.1 C++

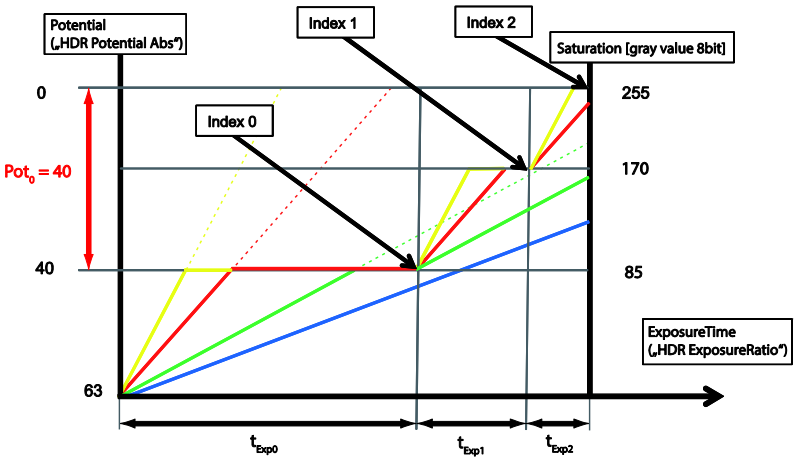
This chapter describes setting of *HDREnableTriggerAutoMode = False & ExposureMode = TriggerControlled* in C++.

```
SystemList
Open a System
Get the InterfaceList and fill it
Open an Interface
Get the DeviceList and fill it
Open a Device


//Device Parameter Setup

if(pDevice->GetRemoteNode("ExposureMode")->GetValue() == "Timed")

{

    pDevice->GetRemoteNode("TriggerMode")->SetString("On");

}

std::cout << " TriggerMode:"
          << pDevice->GetRemoteNode("TriggerMode")->GetValue()
          << std::endl;


pDevice->GetRemoteNode("TriggerSource")->SetString("Line0");

std::cout << "TriggerSource:"
          << pDevice->GetRemoteNode("TriggerSource")->GetValue()
          << std::endl;


pDevice->GetRemoteNode("TriggerActivation")->SetString("RisingEdge");

std::cout << "TriggerActivation: "
          << pDevice->GetRemoteNode("TriggerActivation")->GetValue()
          << std::endl;
```

```
pDevice->GetRemoteNode("ExposureMode")->SetString("TriggerControlled");

std::cout << "ExposureMode:"
          << pDevice->GetRemoteNode("ExposureMode")->GetValue()
          << std::endl;


//HDR parameter change

std::cout << "HDR parameter change" << std::endl;

pDevice->GetRemoteNode("HDREnable")->SetBool(true);

std::cout << "HDREnable: "
          << pDevice->GetRemoteNode("HDREnable")->GetBool()
          << std::endl;


//only HXG

pDevice->GetRemoteNode("HDREnableTriggerAutoMode")->SetBool(false);

std::cout << "HDREnableTriggerAutoMode: "
          << pDevice->GetRemoteNode("HDREnableTriggerAutoMode")->GetBool()
          << std::endl;


pDevice->GetRemoteNode("HDRIndex")->SetInt(0);

std::cout << "HDRIndex: "
          << pDevice->GetRemoteNode("HDRIndex")->GetInt()
          << std::endl;


//pDevice->GetRemoteNode("HDRExposureRatio")->SetInt(185);

//std::cout << "HDRExposureRatio: "
            << pDevice->GetRemoteNode("HDRExposureRatio")->GetInt()
            << std::endl;

//std::cout << "HDRExposureRatioPercent: "
            << pDevice->GetRemoteNode("HDRExposureRatioPercent")->GetDouble()
            << std::endl;


pDevice->GetRemoteNode("HDRPotentialAbs")->SetInt(40); //Pot_0

std::cout << "HDRPotentialAbs: "
          << pDevice->GetRemoteNode("HDRPotentialAbs")->GetInt()
          << std::endl;
```

Notice

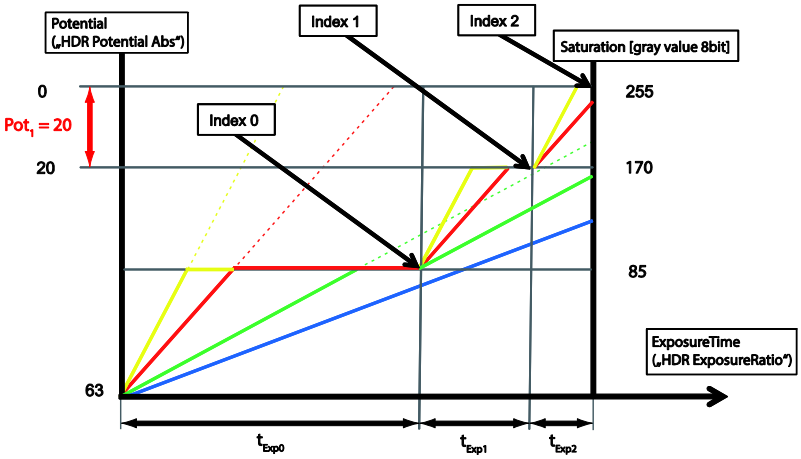
If `HDREnableTriggerAutoMode = False`, `HDRExposureRatio` and `HDRExposureRatioPercent` will not be effected.




```
pDevice->GetRemoteNode("HDRIndex")->SetInt(1);

std::cout << "HDRIndex: "
          << pDevice->GetRemoteNode("HDRIndex")->GetInt()
          << std::endl;


//pDevice->GetRemoteNode("HDRExposureRatio")->SetInt(45); //texp1

//std::cout << "   HDRExposureRatio: "
            << pDevice->GetRemoteNode("HDRExposureRatio")->GetInt()
            << std::endl;

//std::cout << "HDRExposureRatioPercent: "
            << pDevice->GetRemoteNode("HDRExposureRatioPercent")->GetDouble()
            << std::endl;


pDevice->GetRemoteNode("HDRPotentialAbs")->SetInt(20); //Pot_1

std::cout << "HDRPotentialAbs: "
          << pDevice->GetRemoteNode("HDRPotentialAbs")->GetInt()
          << std::endl;
```

Notice

If `HDREnableTriggerAutoMode = False`, `HDRExposureRatio` and `HDRExposureRatioPercent` will not be effected.




```
Get DataStreamList and fill it
Open a Data Stream
Create the BufferList an allocate Buffer memory
Allocate Image Buffer to the DataStream
Start Camera and fill the Image Buffer
Releasing the resources
```

```
TriggerMode: On

TriggerSource: Line0

TriggerActivation: RisingEdge

ExposureMode: TriggerControlled

HDR parameter change
HDREnable: True
HDREnableTriggerAutoMode: False
HDRIndex: 0
HDRPotentialAbs: 40
HDRIndex: 1
HDRPotentialAbs: 20
```

Console Output (C++)

2.2.4.2 C#

This chapter describes setting of *HDREnableTriggerAutoMode = False & ExposureMode = TriggerControlled* in C#.

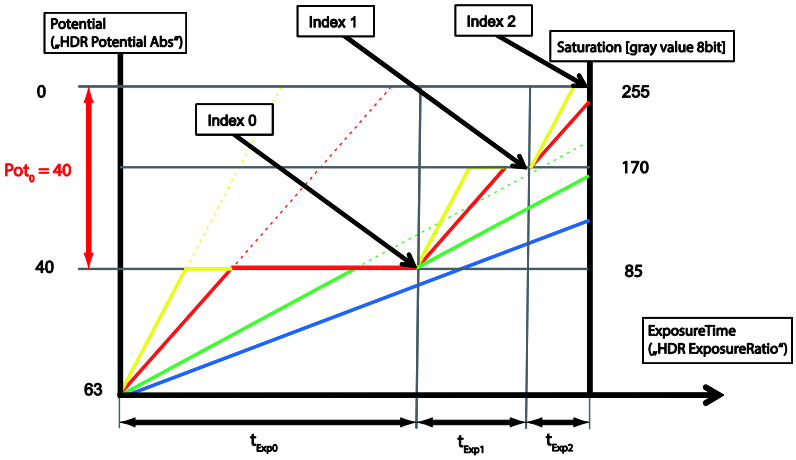
```
SystemList
Open a System
Get the InterfaceList and fill it
Open an Interface
Get the DeviceList and fill it
Open a Device
```

```
//Device Parameter Setup

if ((string)mDevice.RemoteNodeList["ExposureMode"].Value == "Timed")

{

    mDevice.RemoteNodeList["TriggerMode"].Value = "On";

}

System.Console.Write("TriggerMode: {0}\n\n",
   (string)mDevice.RemoteNodeList["TriggerMode"].Value);


mDevice.RemoteNodeList["TriggerSource"].Value = "Line0";

System.Console.Write("TriggerSource: {0}\n\n",
   (string)mDevice.RemoteNodeList["TriggerSource"].Value);


mDevice.RemoteNodeList["TriggerActivation"].Value = "RisingEdge";

System.Console.Write("TriggerActivation: {0}\n\n",
   (string)mDevice.RemoteNodeList["TriggerActivation"].Value);


mDevice.RemoteNodeList["ExposureMode"].Value = "TriggerControlled";

System.Console.Write("ExposureMode: {0}\n\n",
 (string)mDevice.RemoteNodeList["ExposureMode"].Value);


//HDR parameter change

System.Console.Write("HDR parameter change\n");

mDevice.RemoteNodeList["HDREnable"].Value = true;

System.Console.Write("HDREnable: {0}\n",
   (bool)mDevice.RemoteNodeList["HDREnable"].Value);
```

```
//only HXG

mDevice.RemoteNodeList["HDREnableTriggerAutoMode"].Value = false;

System.Console.Write("HDREnableTriggerAutoMode: {0}\n",
   (bool)mDevice.RemoteNodeList["HDREnableTriggerAutoMode"].Value);

mDevice.RemoteNodeList["HDRIndex"].Value = (long)0;

System.Console.Write("  HDRIndex: {0}\n",
   (long)mDevice.RemoteNodeList["HDRIndex"].Value);


//mDevice.RemoteNodeList["HDRExposureRatio"].Value = (long)185;

//System.Console.Write("HDRExposureRatio: {0}\n",
   (long)mDevice.RemoteNodeList["HDRExposureRatio"].Value);

//System.Console.Write("HDRExposureRatioPercent: {0:F2}\n",
   (double)mDevice.RemoteNodeList["HDRExposureRatioPercent"].Value);


mDevice.RemoteNodeList["HDRPotentialAbs"].Value = (long)40; //Pot_0

System.Console.Write("HDRPotentialAbs: {0}\n",
   (long)mDevice.RemoteNodeList["HDRPotentialAbs"].Value);
```

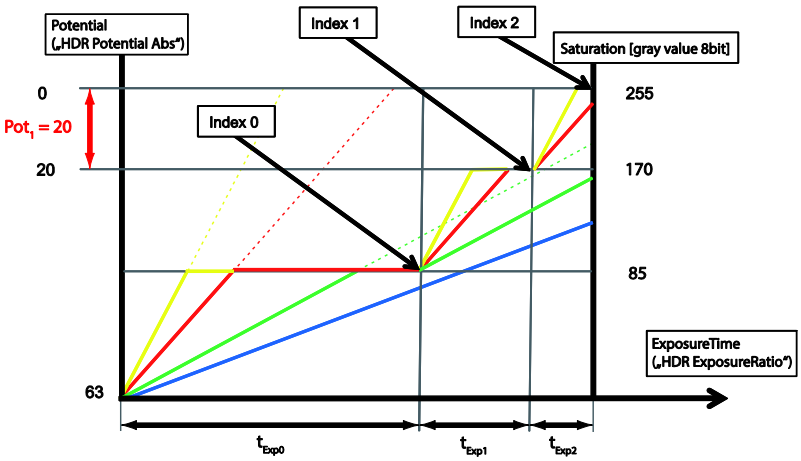
Notice

If `HDREnableTriggerAutoMode = False`, `HDRExposureRatio` and `HDRExposureRatioPercent` will not be effected.




```
mDevice.RemoteNodeList["HDRIndex"].Value = (long)1;

System.Console.Write("HDRIndex: {0}\n",
   (long)mDevice.RemoteNodeList["HDRIndex"].Value);

//mDevice.RemoteNodeList["HDRExposureRatio"].Value = (long)45;
```

```
//System.Console.Write("HDRExposureRatio: {0}\n",
   (long)mDevice.RemoteNodeList["HDRExposureRatio"].Value);

//System.Console.Write("HDRExposureRatioPercent: {0:F2}\n",
   (double)mDevice.RemoteNodeList["HDRExposureRatioPercent"].Value);

mDevice.RemoteNodeList["HDRPotentialAbs"].Value = (long)20; //Pot_0

System.Console.Write("HDRPotentialAbs: {0}\n",
   (long)mDevice.RemoteNodeList["HDRPotentialAbs"].Value);

System.Console.Write("\n")
```




```
Get DataStreamList and fill it
Open a Data Stream
Create the BufferList an allocate Buffer memory
Allocate Image Buffer to the DataStream
Start Camera and fill the Image Buffer
Releasing the resources
```

```
TriggerMode: On

TriggerSource: Line0

TriggerActivation: RisingEdge

ExposureMode: TriggerControlled

HDR parameter change
HDREnable: True
HDREnableTriggerAutoMode: False
HDRIndex: 0
HDRPotentialAbs: 40
HDRIndex: 1
HDRPotentialAbs: 20
```

Console Output (C#)

3 Support

In the case of any questions or for troubleshooting please contact our support team.

Worldwide

Baumer Optronic GmbH
Badstrasse 30 · DE-01454 Radeberg
Phone +49 3528 4386 845
support.cameras@baumer.com

4 Disclaimer

All other product and company names mentioned are trademarks or registered trademarks of their respective owners.

Baumer
Passion for Sensors

Baumer Optronic GmbH
Badstrasse 30 · DE-01454 Radeberg
Phone +49 3528 4386 0 · Fax +49 3528 4386 86
sales@baumeroptronic.com · www.baumer.com