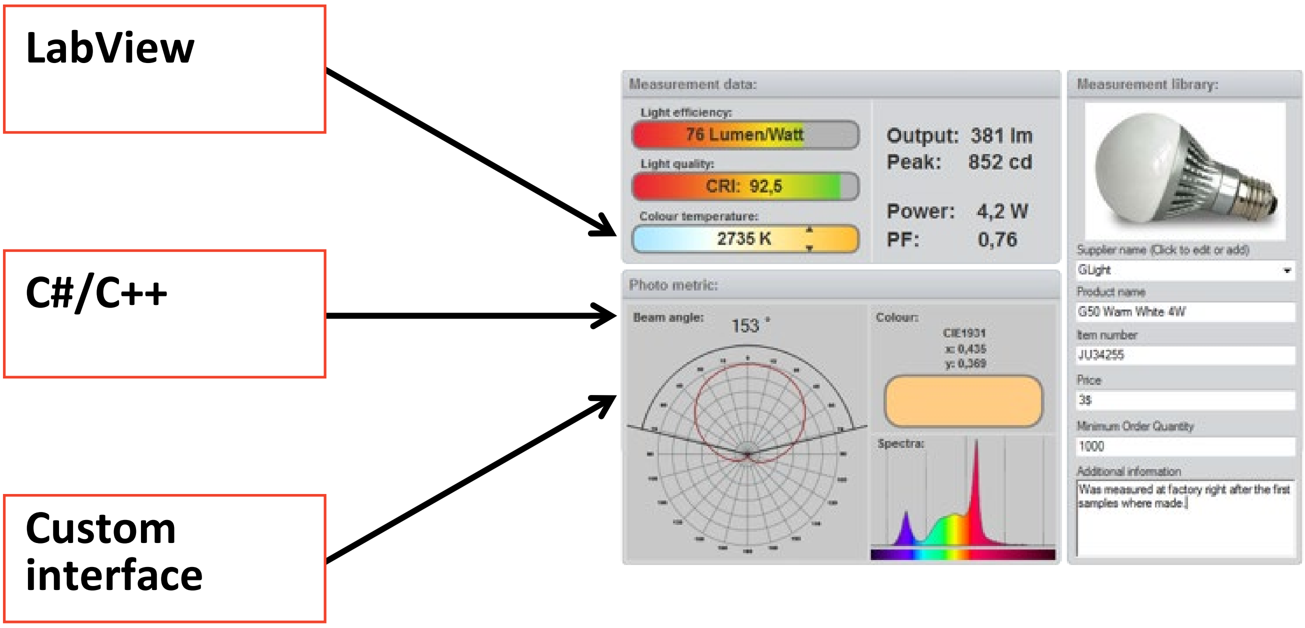
VISO SYSTEMS

.NET DLL Interface to Light Inspector software

Revision: 04-12-2024



LabView

C#/C++

Custom interface

Measurement data:

Light efficiency: 76 Lumen/Watt

Light quality: CRI: 92,5

Colour temperature: 2735 K

Output: 381 lm
Peak: 852 cd

Power: 4,2 W
PF: 0,76

Measurement library:

Supplier name (Click to edit or add): GLight

Product name: G50 Warm White 4W

Item number: JU34255

Price: 3$

Minimum Order Quantity: 1000

Additional information: Was measured at factory right after the first samples where made.

Photo metric:

Beam angle: 153 °

Colour: CIE1931
x: 0,435
y: 0,369

Spectra:

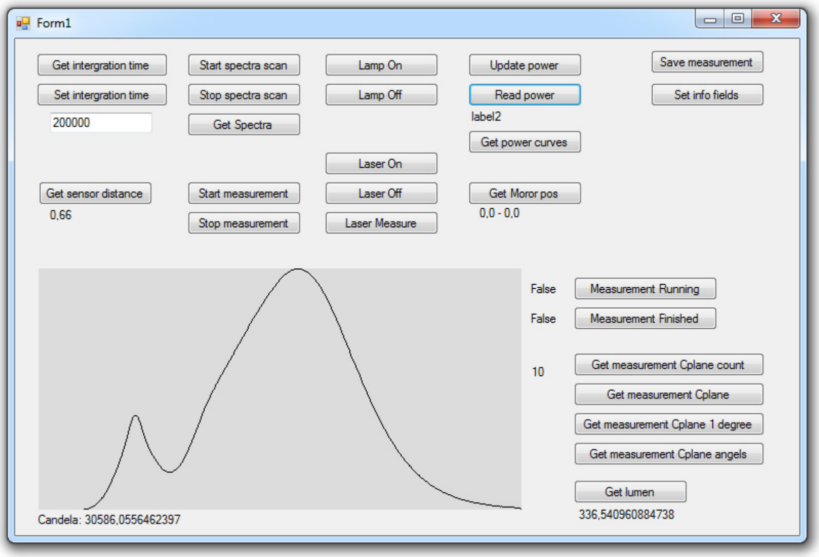
Introduction

The DLL interface for the Viso Light Inspector software allows you to control the light measurement system manually or to do your own calculation of the results measured by the system.

The .NET dll is located in the install folder of the Light Inspector software and has the name “Interfacing.dll”.

An open-source demo c# application can be downloaded using the link below.

http://www.visosystems.com/software/Interface dll demo v1.00.zip



```
200000
0,66
0,0 - 0,0
False
False
10
Candela: 30586,0556462397
336,540960884738
```

Functions

Goniospectrometer

```
public int GetIntegrationTime()

public void OverrideIntegrationTime(int IntegrationTime)

public void ClearOverrideIntegrationTime()

public void StartSpectrumScan()

public void StopSpectrumScan()

public bool NewSpectrumAvaileble()

public double[] GetSpectra_Scaled()

public void GetSpectraAndPhotometric(out double[]
Spectra_watt_sr_nm, out double candela, out double
cie1931_x,out double cie1931_y,out double colorTemperature,
out double CRI, out double[] R_values, out double SDCM)
```

```
public void StartMeasurement(int CplaneCount)

public void StartMeasurementWithWarmup(int CplaneCount, int
warmupOption = 9)

//For more options look in the dropdown, in the warmup
window.

public void GetWarmupCurves(out double[] intensity, out
double[] power, out double[] voltage)

public double GetWarmupTime()

public void StopMeasurement()

public double GetSensorDistance()

public double GetCandelaValue()

public double GetLumenValue()

public void SetSensorDistance(double distance)

public void GetGonioPosition(out double position,out double
position_cplane)

public void SetGonioPosition(double position, double
position_cplane)

public void LampOn()

public void LampOff()

public void UpdatePower()

public void LaserOn()

public void LaserOff()

public void LaserMeasure()

public void MakeAutoIntegrationAndAmbient()

public void GetPower(out double RMS_Voltage, out double
RMS_Current, out double VA, out double Power, out double
PowerFactor, out double Frequency)

public void GetPowerCurves(out double[] Voltage, out
double[] Current)

public double GetPower()

public bool MeasurementRunning()

public bool MeasurementFinished()

public int GetMeasurementCPlaneCount()

public double[] GetMeasurementCPlaneAngels()

public int GetMeasurementCPlaneRAW(int CplaneIndex,out
double[] angl,out double[] candela)
```

```
public int GetMeasurementCPlane1DegreeStep(int CplaneIndex,
out double[] angl, out double[] candela)

public void SaveMeasurement(string MeasurementName)

public bool MeasurementExist(string MeasurementName)

public void SetInfoFields(string MeasurementName, string
ItemNumber, string Price, string MinimumOrder, string
AdditionalInfo)

public void SetInfoFields(string MeasurementName, string
ItemNumber, string Price, string MinimumOrder, string[]
AdditionalInfo)

public void SetPowerContorlAuto()

public void SetPowerContorlManual()

public void SetPowerContorlDisabled()

public bool IsTurnOnLightOpen()

public bool IsTurnOffLightOpen()

public void ConfirmLightOn()

public void ConfirmLightOff()
```

LabFlicker

```
public void OpenFlickerWindow()

public void CloseFlickerWindow()

public bool GetIsFlickerConnected()

public bool GetIsFlickerMeasuringInProgress()

public void FlickerPressButton()

public void FlickerStartSVMMeasurement()

public void FlickerStartPstLMMeasurement()

public double GetFlickerIndex()

public double GetFlickerPercent()

public double GetFlickerFrequency()

public double GetFlickerSVM()

public double GetFlickerPstLM()

public bool GetFlickerIsIntensityOK()

public double GetFlickerIntensityValue()

public double[] GetFlickerRAWSVM()

public double[] GetFlickerRAWPstLM()
public double[] GetFlickerJA8() //Values 0: 40Hz, 1: 90Hz, 2: 200Hz, 3: 400Hz, 4: 1000Hz
```

Export

```
public void SaveToXlsx(string filename, int itemNumber)

//filename: Full path and file name

//itemNumber: 1: Photometric data, 2: Photometric, 3: Spectras for all angles, 4: Flicker, 5: Temperature log

public void SaveToCSV(string filename, int itemNumber)

//filename: Full path and file name

//itemNumber: 1: Photometric data, 2: Photometric, 3: Spectra for all angles, 4: Flicker, 5: Flicker PstLM, 6: Temperature log
```

Labarazzi

```
public void TLASelectWaveform string waveform_name,float lumen) – calling first name match (custom waveforms prioritized over built-in waveforms)

public void TLAStartLED()

public void TLAStopLED()
```

At Viso Systems, we design, develop and manufacture OEM- and customer-specific goniophotometer solutions. Our mission is to support customers with powerful, yet easy-to-use control and measurement solutions. Products are developed and manufactured in Copenhagen, Denmark.

VISO SYSTEMS

Light measurement made easy