

The information contained in this documentation is the property of MAZeT. Photocopying or otherwise reproducing any part of the catalog, whether electronically or mechanically, is prohibited, except where the express permission of MAZeT GmbH has been obtained. In general, all company and brand names, as well as the names of individual products, are protected by brand, patent or product law.

**VERSION AMENDMENTS**

NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

**Software Description**

**JENCOLOR Library  
JenColorDLL**

**Description of the library JenColorDLL.**

**Calibration, measurement and color metrics for True Color Sensors.**

Revision: 1.00

**Table of contents**

<b>1 JENCOLOR Library: JenColorDLL</b>	<b>2</b>
1.1 Introduction .....	2
1.2 Todo .....	2
1.3 Library initialisation .....	3
1.4 Sensor calibration.....	3
1.5 Measurements.....	3
<b>2 JENCOLORLibrary File Documentation</b>	<b>4</b>
2.1 JenColorDLL.h File Reference.....	4
<b>3 JENCOLORLibrary Page Documentation</b>	<b>27</b>
3.1 Todo List .....	27

<b>MAZeT GmbH</b> Sales Department Göschwitzer Straße 32 07745 Jena / Germany Tel.: +49 3641 2809-0 Fax: +49 3641 2809-12 Email: sales@MAZeT.de Url: http://www.MAZeT.de	<b>Acknowledgement</b>	<b>Date</b>	<b>MAZeT GmbH</b>	
	Created:	11.12.2007	Status: Valid	
	Checked:	11.12.2007		
	Released:	29.07.2008	DOC. NO.: DB08242e	Page 1 of 30

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

# 1 JENCOLOR Library: JenColorDLL

## 1.1 Introduction

The following document describes the use of the library JenColorDLL. This library supports the calibration, measurements and color metrics with a MTCS true color sensor of MAZeT.

Copy the three files "JenColorDLL.dll", "JenColorDLL.lib" and "JenColorDLL.h" in your project folder. To use the library include "JenColorDLL.h" and "JenColorDLL.lib" in your project.

The "JenColorDLL" uses function pointer as callback functions to adjust it to your application. Thus, you can use any bus interface to read out the three sensor channels or save system data after calibration. If any of these functions is not needed (like turn on/off lighting when measuring self-illuminating targets) just leave them unprogrammed.

## 1.2 Todo

The following function pointer have to be initialised and programmed for your application (further information in the corresponding description and source code):

a) **TMva\_LoadTargetXYZ()** (p. 13): Called from **Mva\_StartCalibration()** (p. 20).

b) **TMva\_SetTargetColor()** (p. 15): Called from **Mva\_StartCalibration()** (p. 20).

c) **TMva\_SaveSettings()** (p. 14): Needs to be called after calibration.

d) **TMva\_LoadSettings()** (p. 13): Called from **Mdi\_InitLibrary()** (p. 19).

e) **TMdi\_InitGlobals()** (p. 9): Called from **Mdi\_InitLibrary()** (p. 19).

f) **TMdi\_StartADCReturnInt()** (p. 12):

Called from **Mva\_StartCalibration()** (p. 20) and **Mva\_GetAdjustedValues()** (p. 19).

g) **TMdi\_StartADCReturnFloat()** (p. 11):

Called from **Mva\_StartCalibration()** (p. 20) and **Mva\_GetAdjustedValues()** (p. 19).

h) **TMdi\_MeasuringError()** (p. 10):

Called from **Mva\_StartCalibration()** (p. 20) and **Mva\_GetAdjustedValues()** (p. 19).

i) **TMdi\_LightingOn()** (p. 10):

Called from **Mva\_StartCalibration()** (p. 20) and **Mva\_GetAdjustedValues()** (p. 19).

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

**j) TMdi\_LightingOff()** (p. 9):

Called from **Mva\_StartCalibration()** (p. 20) and **Mva\_GetAdjustedValues()** (p. 19).

If f) is used, g) is left unprogrammed.

If g) is used, f) is left unprogrammed.

If no lighting exists, i) and j) are left unprogrammed.

If no calibration is executed, a), b) and c) are left unprogrammed.

### 1.3 Library initialisation

1. step: **SetCallbackProcAddr()** (p. 20) explained on referenced page.

2. step: **Mdi\_InitLibrary()** (p. 19) explained on referenced page.

### 1.4 Sensor calibration

1. step: **Mva\_StartCalibration()** (p. 20) explained on referenced page.

2. step: **TMva\_SaveSettings()** (p. 14) explained on referenced page.

### 1.5 Measurements

1. step: **Mva\_GetAdjustedValues()** (p. 19) explained on referenced page.

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

## 2 JENCOLORLibrary File Documentation

### 2.1 JenColorDLL.h File Reference

Header-file of "JenColorDLL.dll".

#### Defines

- #define **COLORDLL\_API** \_\_declspec(dllimport)  
*Definition to import the JenColorDLL. Do not use COLORDLL\_API as a definition in other functions.*

#### Typedefs

- typedef void(\*) **TMva\_LoadTargetXYZ** (float pTargetValues[ ])  
*Function pointer on a callback function to load the XYZ target values during calibration.*
- typedef unsigned char(\*) **TMva\_SetTargetColor** (unsigned char)  
*Function pointer on a callback function to set the passed target number during calibration.*
- typedef unsigned char(\*) **TMva\_SaveSettings** (void)  
*Function pointer on a callback function for the non-volatile storage of the calibration and sensor data.*
- typedef unsigned char(\*) **TMva\_LoadSettings** (void)  
*Function pointer on a callback function to load and initialise the calibration and sensor data.*
- typedef unsigned char(\*) **TMdi\_InitGlobals** (unsigned char \*NumberOfTC, unsigned char \*NumberOfMC, unsigned char \*NumberOfOtherSens, unsigned char \*NumberOfTargets, unsigned char \*NumberOfInterpolation)  
*Function pointer on a callback function to initialise the global variables of the JenColorDLL.*

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

- typedef void(\*) **TMdi\_StartADCReturnInt** (unsigned char MeasureSensor, unsigned int pADC[])  
*Function pointer on a callback function to start AD conversation and return integer values.*
- typedef void(\*) **TMdi\_StartADCReturnFloat** (unsigned char MeasureSensor, float pADC[])  
*Function pointer on a callback function to start AD conversation and return float values.*
- typedef void(\*) **TMdi\_MeasuringError** (unsigned char Sensor, unsigned char Channel, unsigned char Error)  
*Function pointer on a callback function for an error handling routine during measurements.*
- typedef void(\*) **TMdi\_LightingOff** (unsigned char MeasureSensor)  
*Function pointer on a callback function to turn off the lighting.*
- typedef void(\*) **TMdi\_LightingOn** (unsigned char MeasureSensor)  
*Function pointer on a callback function to turn on the lighting.*

## Enumerations

- enum **PROGS** {  
**MVA\_SAVESETTINGS = 0, MVA\_LOADSETTINGS, MVA\_SETTARGETCOLOR, MVA\_LOADTARGETXYZ,**  
**MDI\_INITGLOBALS, MDI\_MEASURINGERROR, MDI\_LIGHTINGON, MDI\_LIGHTINGOFF,**  
**MDI\_STARTADCRETURNTINT, MDI\_STARTADCRETURNFLOAT }**
- Integer constants used to allocate the function pointer. Do not change the constants!*

## Functions

- COLORDLL\_API int **SetCallbackProcAddr** (long IProcAdr, int iIndex)  
*Initialisation of the function pointer for the callback functions.*

NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

- COLORDLL\_API unsigned char **Mdi\_InitLibrary** (unsigned char \*pError)  
*Initialisation of the library.*
- COLORDLL\_API unsigned char **Mva\_StartCalibration** (unsigned char Selected-Sensor)  
*Starts calibration of true color sensors.*
- COLORDLL\_API unsigned char **Mva\_GetAdjustedValues** (unsigned char SelectedSensor)  
*Starts measuring and value adjustment of sensors.*
- void COLORDLL\_API **Mva\_GetVersion** (char \*pcBuf)  
*Readout of version information Mva (Value Adjustment).*
- COLORDLL\_API void **Mcm\_GetVersion** (char \*pcBuf)  
*Readout of version information Mcm (Color Metrics).*
- COLORDLL\_API void **Mcm\_XYZtoxyY** (float pdXYZ[], float pdxyY[])  
*Conversion from XYZ to xyY.*
- COLORDLL\_API void **Mcm\_XYZtoLuv** (float pdXYZ[], float pdLUV[])  
*Conversion from XYZ to L\*u\*v'.*
- COLORDLL\_API void **Mcm\_XYZtoLab** (float XYZ[], float pdLab[], float XnYnZn[])  
*Conversion from XYZ to L\*a\*b\*.*
- COLORDLL\_API void **Mcm\_RGBtoXYZ** (float pdRGB[], float Matrix[], float pdXYZ[])  
*Conversion from RGB to XYZ.*
- COLORDLL\_API float **Mcm\_deltaEab** (float LabA[], float LabB[])  
*Calculates the color difference DeltaE (CIELAB) between two colors L\*a\*b\*(A) and L\*a\*b\*(B).*
- COLORDLL\_API void **Mcm\_XYZtoRGB** (float XYZ[], float fMatrix[], float pdRGB[])

NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

*Conversion from XYZ to RGB.*

## Variables

- **TMva\_LoadTargetXYZ PMva\_LoadTargetXYZ**  
*Variable for imported function (internal use).*
- **TMva\_SetTargetColor PMva\_SetTargetColor**  
*Variable for imported function (internal use).*
- **TMva\_SaveSettings PMva\_SaveSettings**  
*Variable for imported function (internal use).*
- **TMva\_LoadSettings PMva\_LoadSettings**  
*Variable for imported function (internal use).*
- **TMdi\_InitGlobals PMdi\_InitGlobals**  
*Variable for imported function (internal use).*
- **TMdi\_StartADCReturnInt PMdi\_StartADCReturnInt**  
*Variable for imported function (internal use).*
- **TMdi\_StartADCReturnFloat PMdi\_StartADCReturnFloat**  
*Variable for imported function (internal use).*
- **TMdi\_MeasuringError PMdi\_MeasuringError**  
*Variable for imported function (internal use).*
- **TMdi\_LightingOn PMdi\_LightingOn**  
*Variable for imported function (internal use).*
- **TMdi\_LightingOff PMdi\_LightingOff**  
*Variable for imported function (internal use).*
- **COLORDLL\_API unsigned int glibMeasure [ ]**  
*Contains the number of measurements each sensor for averaging.*

NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

- COLORDLL\_API unsigned char **glibSensor** []  
*Contains information about the sensor like the measure sequence, resolution and lighting.*
- COLORDLL\_API signed char **glibScale** []  
*Contains the scalefactor of the sensors, calculated during calibration.*
- COLORDLL\_API float **glibAdjust** []  
*Contains the values for the adjustment of the sensors.*
- COLORDLL\_API float **glibResult** []  
*Result array of the library JenColorDLL. Contains the sensor results after a measurement.*

### 2.1.1 Detailed Description

Header-file of "JenColorDLL.dll".

Project : JENCOLOR Library

Package : JenColorDLL

Company : MAZeT GmbH; Goeschwitzer Strasse 32; D-07745 Jena

Version History:

V 0.10 11.12.2007 First Revision

Definition in file **JenColorDLL.h**.

### 2.1.2 Define Documentation

#### 2.1.2.1 #define COLORDLL\_API \_\_declspec(dllimport)

Definition to import the JenColorDLL. Do not use COLORDLL\_API as a definition in other functions.

Definition at line 326 of file JenColorDLL.h.

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

### 2.1.3 Typedef Documentation

#### 2.1.3.1 typedef unsigned char(\* ) TMDi\_InitGlobals(unsigned char \*NumberOfTC, unsigned char \*NumberOfMC, unsigned char \*NumberOfOtherSens, unsigned char \*NumberOfTargets, unsigned char \*NumberOfInterpolation)

Function pointer on a callback function to initialise the global variables of the JenColorDLL.

The return value decides which function is called to read out the ADC values. With pCol2StartADCReturnInt() an averaging is done with a number of measurements saved in glibMeasure. With pCol2StartADCReturnFloat() the value in glibMeasure is ignored and the user can program his own signal processing like FIR filter.

#### Todo

Change the status value for function call during measurement, number of true color sensors and calibration targets to your applications need.

#### Parameters:

*None.*

#### Return values:

**NumberOfTC** Number of true color sensors (min = 1, max = 10).

**aNotUsed** Not used yet.

**bNotUsed** Not used yet.

**NumberOfTargets** Number of calibration targets (min = 3, max = 100).

**cNotUsed** Not used yet.

#### Returns:

Status value for function call during measurement.

= 0 Start ADC with pCol2StartADCReturnInt()

!= 0 Start ADC with pCol2StartADCReturnFloat()

Definition at line 210 of file JenColorDLL.h.

#### 2.1.3.2 typedef void(\* ) TMDi\_LightingOff(unsigned char MeasureSensor)

Function pointer on a callback function to turn off the lighting.

If only one lighting exists, it can be turned off without checking parameter "MeasureSensor".

If no lighting exists, leave the function unprogrammed.

NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

**Todo**

If your system uses lighting, write a routine to turn off the light.

**Parameters:**

**MeasureSensor** =1..10 Turn off lighting of passed parameter MeasureSensor.

**Return values:**

**None.**

**Returns:**

None.

Definition at line 304 of file JenColorDLL.h.

**2.1.3.3 typedef void(\* ) TMdi\_LightingOn(unsigned char MeasureSensor)**

Function pointer on a callback function to turn on the lighting.

If only one lighting exists, it can be turned on without checking paramter "MeasureSensor".

If no lighting exists, leave the function unprogrammed.

**Todo**

If your system uses lighting, write a routine to turn on the light.

**Parameters:**

**MeasureSensor** =1..10 Turn on lighting of passed parameter MeasureSensor

**Return values:**

**None.**

**Returns:**

None.

Definition at line 321 of file JenColorDLL.h.

**2.1.3.4 typedef void(\* ) TMdi\_MeasuringError(unsigned char Sensor, unsigned char Channel, unsigned char Error)**

Function pointer on a callback function for an error handling routine during measurements.

This routine is called, if a sensor channel caused a zero signal or a signal overflow (ADC value reaches maximum value minus 1% tolerance) after a AD conversation.

NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

**Todo**

Write your own error handling routine if a sensor causes a zero signal or signal overflow.

**Parameters:**

**Sensor** Mismeasuring of passed sensor.

**Channel** Mismeasuring of passed channel.

**Error** =1 zero signal  
=2 overamplify

**Return values:**

**None.**

**Returns:**

None.

Definition at line 287 of file JenColorDLL.h.

**2.1.3.5 typedef void(\* ) TMdi\_StartADCReturnFloat(unsigned char MeasureSensor, float pADC[ ])**

Function pointer on a callback function to start AD conversation and return float values.

If only one sensor is connected, it can be read out without checking parameter "MeasureSensor".

Stick to the following sequence, when measuring a true color sensor:

pAavgADC[0] = Channel Red

pAavgADC[1] = Channel Green

pAavgADC[2] = Channel Blue

**Todo**

Write your own AD conversation or interface functions to read out a true color sensor and return float values. With this function the internal averaging cannot be used.

**Warning:**

ATTENTION FOR OVERFLOW:

Arraysizes cannot be checked inside the JenColorDLL!

max. arraysize of pAavgADC[] = Number of channels of the measured sensor.

i.e. True Color sensor: max. arraysize = 3

**Parameters:**

**MeasureSensor** =1..10 Measuring sensor.

NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

**Return values:**

***pAavgADC[]*** Pointer on float ADC-values of the measured sensor.

**Returns:**

None.

Definition at line 266 of file JenColorDLL.h.

**2.1.3.6 typedef void(\* ) TMDi\_StartADCReturnInt(unsigned char MeasureSensor, unsigned int pADC[])**

Function pointer on a callback function to start AD conversation and return integer values.

If only one sensor is connected, it can be read out without checking parameter "MeasureSensor".

Stick to the following sequence, when measuring a true color sensor:

pADC[0] = Channel Red

pADC[1] = Channel Green

pADC[2] = Channel Blue

**Todo**

Write your own AD conversation or interface functions to read out a true color sensor and return integer values. With this function the internal averaging can be used.

**Warning:**

ATTENTION FOR OVERFLOW:

Arraysizes cannot be checked inside the JenColorDLL!

max. arraysize of pAavgADC[] = Number of channels of the measured sensor.

i.e. True Color sensor: max. arraysize = 3

**Parameters:**

***MeasureSensor*** =1..10 Measuring sensor.

**Return values:**

***pADC[]*** Pointer on integer ADC-values of the measured sensor.

**Returns:**

None.

Definition at line 240 of file JenColorDLL.h.

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

### 2.1.3.7 typedef unsigned char(\*) TMva\_LoadSettings(void)

Function pointer on a callback function to load and initialise the calibration and sensor data.

The calibration and sensor data saved with pCol2SaveSettings() are loaded for the initialisation of the JenColorDLL.

#### Todo

Reload the arrays glibAdjust, glibMeasure, glibScale and glibSensor for each sensor from a non-volatile storage. (Saved before in **TMva\_SaveSettings()** (p. 14).)

#### Warning:

ATTENTION FOR OVERFLOW:

Arraysizes cannot be checked inside the JenColorDLL!

Arraysize glibMeasure = Number of true color sensors.

Arraysize **glibScale()** (p. 23) = Number of true color sensors.

Arraysize **glibSensor()** (p. 24) = 3x Number of true color sensors.

Arraysize **glibAdjust()** (p. 22) = 15x Number of true color sensors.

#### Parameters:

*None.*

#### Return values:

**glibAdjust()** (p. 22) Contains offset values, black-white offset values and calibration matrices of the sensors.

**glibMeasure()** (p. 22) Contains the number of measurements for the sensors.

**glibScale()** (p. 23) Contains scale factors of the sensors.

**glibSensor()** (p. 24) Contains the measure sequence, bit resolution and system lighting of the sensors.

#### Returns:

iErr Error code defined by user. This value is not checked in the JenColorDLL.

User can check this value, when it is returned from Mdi\_InitLibrary(pError).

=0 Settings loaded successfully.

!=0 Userdefined error.

Definition at line 184 of file JenColorDLL.h.

### 2.1.3.8 typedef void(\*) TMva\_LoadTargetXYZ(float pTargetValues[ ])

Function pointer on a callback function to load the XYZ target values during calibration.

NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

### Todo

The XYZ target values have to be adjusted on the application! Measure your calibration targets with a calibrated spectrometer and insert the XYZ target values in this function.

### Warning:

ATTENTION FOR OVERFLOW:

Arraysizes cannot be checked inside the JenColorDLL!

The maximum size of pTargetValues[] is:  $3 * \text{NumberOfTargets} - 1$   
i.e.: 24 Targets: maximum size =  $3 * 24 - 1 = 71$

### Parameters:

*None.*

### Return values:

**pTargetValues** Pointer on array for the XYZ target values. The order of the values is pTargetValues[] = {X1, Y1, Z1, X2, Y2, Z2, ...}.

### Returns:

None.

Definition at line 100 of file JenColorDLL.h.

### 2.1.3.9 typedef unsigned char(\*) TMva\_SaveSettings(void)

Function pointer on a callback function for the non-volatile storage of the calibration and sensor data.

The calibration and sensor data have to be stored non-volatile. During the library initialisation **Mdi\_InitLibrary()** (p. 19) these values are reloaded and the sensor has not to be calibrated again.

### Todo

Save the arrays glibAdjust, glibMeasure, glibScale and glibSensor for each sensor on a non-volatile storage.

### Warning:

ATTENTION FOR OVERFLOW:

Arraysizes cannot be checked inside the JenColorDLL!

Arraysize glibMeasure = Number of true color sensors.

Arraysize **glibScale()** (p. 23) = Number of true color sensors.

Arraysize **glibSensor()** (p. 24) = 3x Number of true color sensors.

Arraysize **glibAdjust()** (p. 22) = 15x Number of true color sensors.

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

**Parameters:**

***glibAdjust()*** (p. 22) Contains offset values, black-white offset values and calibration matrices of the sensors.

***glibMeasure()*** (p. 22) Contains the number of measurements for the sensors.

***glibScale()*** (p. 23) Contains scale factors of the sensors. ABCDEF

***glibSensor()*** (p. 24) Contains the measure sequence, bit resolution and system lighting of the sensors.

**Return values:**

***None.***

**Returns:**

iErr Error code defined by user. This value is not checked in the JenColorDLL.

=0 Settings saved successfully.

!=0 Userdefined error.

Definition at line 151 of file JenColorDLL.h.

**2.1.3.10 typedef unsigned char(\*) TMva\_SetTargetColor(unsigned char)**

Function pointer on a callback function to set the passed target number during calibration.

**Todo**

Adjust this function to your application.

i.e. this function can be automated if a multicolor LED or a display is measured. The LED/display can be set automatically to the specific color without asking the user to do this.

**Parameters:**

***TargetNumber*** =1..n Set the target number n.

=0 special case: Offset measurement of sensors.

**Return values:**

***None.***

**Returns:**

NextTarget Variable to stop calibration.

=0 Last target or offset measurement is set.

!=0 Further calibration targets available.

Definition at line 118 of file JenColorDLL.h.

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

## 2.1.4 Enumeration Type Documentation

### 2.1.4.1 enum PROGS

Integer constants used to allocate the function pointer. Do not change the constants!

**Enumerator:**

***MVA\_SAVESETTINGS***  
***MVA\_LOADSETTINGS***  
***MVA\_SETTARGETCOLOR***  
***MVA\_LOADTARGETXYZ***  
***MDI\_INITGLOBALS***  
***MDI\_MEASURINGERROR***  
***MDI\_LIGHTINGON***  
***MDI\_LIGHTINGOFF***  
***MDI\_STARTADCRETURNTINT***  
***MDI\_STARTADCRETURNFLOAT***

Definition at line 372 of file JenColorDLL.h.

## 2.1.5 Function Documentation

### 2.1.5.1 COLORDLL\_API float Mcm\_deltaEab (float *LabA*[], float *LabB*[])

Calculates the color difference DeltaE (CIELAB) between two colors L\*a\*b\*(A) and L\*a\*b\*(B).

**Parameters:**

***LabA*** Pointer on array (arraysize 3) with L\*a\*b\*-values of color A.

***LabB*** Pointer on array (arraysize 3) with L\*a\*b\*-values of color B.

**Return values:**

***None.***

**Returns:**

deltaE between color A and B.

### 2.1.5.2 COLORDLL\_API void Mcm\_GetVersion (char \* *pcBuf*)

Readout of version information Mcm (Color Metrics).

NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

**Parameters:**

*None.*

**Return values:**

*pcBuf* Pointer on array for version information Mcm (arraysize 30 Byte).

**Returns:**

None.

**2.1.5.3 COLORDLL\_API void Mcm\_RGBtoXYZ (float *pdRGB*[], float *Matrix*[], float *pdXYZ*[])**

Conversion from RGB to XYZ.

**Parameters:**

*pdRGB* Pointer on array (arraysize 3) with RGB values.

*Matrix* Pointer on array (arraysize 9) of a transformation matrix. Save a (3,3) transformation matrix line by line into a temporary array to pass it to this function. This matrix is not calculated in the JenColorDLL.

**Return values:**

*pdXYZ* Result, pointer on array (arraysize 3) with XYZ-values.

**Returns:**

None.

**2.1.5.4 COLORDLL\_API void Mcm\_XYZtoLab (float *XYZ*[], float *pdLab*[], float *XnYnZn*[])**

Conversion from XYZ to L\*a\*b\*.

**Parameters:**

*XYZ* Pointer on array (arraysize 3) with XYZ-values.

*XnYnZn* Pointer on array (arraysize 3) with reference white: D65, D55, D75, C, A(2858), A(3200K), none,...

**Return values:**

*pdLab* Result, pointer on array (arraysize 3) with L\*a\*b\*-values.

**Returns:**

None.

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

#### 2.1.5.5 COLORDLL\_API void Mcm\_XYZtoLuv (float *pdXYZ*[], float *pdLUV*[])

Conversion from XYZ to L\*u'v'.

**Parameters:**

*pdXYZ* Pointer on array (arraysize 3) with XYZ-values.

**Return values:**

*pdLUV* Result, pointer on array (arraysize 3) with L\*u'v'-values.

**Returns:**

None.

#### 2.1.5.6 COLORDLL\_API void Mcm\_XYZtoRGB (float *XYZ*[], float *fMatrix*[], float *pdRGB*[])

Conversion from XYZ to RGB.

**Parameters:**

*XYZ* Pointer on array (arraysize 3) with XYZ values.

*fMatrix* Pointer on array (arraysize 9) of a transformation matrix. Save a (3,3) transformation matrix line by line into a temporary array to pass it to this function. This matrix is not calculated in the JenColorDLL.

**Return values:**

*pdRGB* Result, pointer on array (arraysize 3) with RGB-values.

**Returns:**

None.

#### 2.1.5.7 COLORDLL\_API void Mcm\_XYZtoxyY (float *pdXYZ*[], float *pdxyY*[])

Conversion from XYZ to xyY.

**Parameters:**

*pdXYZ* Pointer on array (arraysize 3) with XYZ-values.

**Return values:**

*pdxyY* Result, pointer on array (arraysize 3) with xyY-values.

**Returns:**

None.

NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

### 2.1.5.8 COLORDLL\_API unsigned char Mdi\_InitLibrary (unsigned char \* *pError*)

Initialisation of the library.

**Parameters:**

*None.*

**Return values:**

*pError* user defined error returned from Mva\_LoadSettings()

**Returns:**

- error
- =0 Initialisation ok
- =1 Initialisation not executed
- =2 Max. number of sensors exceeded
- =3 Max. number of true color sensors exceeded
- =4 Max. number of multiple color sensors exceeded
- =5 Max. number of other sensors exceeded
- =6 Max. number of of calibration targets exceeded
- =7 Max. number of interpolation points exceeded
- =8 Max. number of sensor channels exceeded

### 2.1.5.9 COLORDLL\_API unsigned char Mva\_GetAdjustedValues (unsigned char *SelectedSensor*)

Starts measuring and value adjustment of sensors.

After executing result values are available in **glibResult()** (p. 23).

The first function call starts **Mdi\_InitLibrary()** (p. 19), if **Mdi\_InitLibrary()** (p. 19) was not called before.

**Parameters:**

- SelectedSensor* =0 Starts measuring of all sensors
- =1..10 Starts measuring of sensor 1..10

**Return values:**

**glibResult()** (p. 23) Measuring results.

**Returns:**

- Error indicates the status of the measurement.
- =0 Measuring ok
- !=0 Measuring failed. See **Mdi\_InitLibrary()** (p. 19) for details.

NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

#### 2.1.5.10 void COLORDLL\_API Mva\_GetVersion (char \* *pcBuf*)

Readout of version information Mva (Value Adjustment).

**Parameters:**

*None.*

**Return values:**

*pcBuf* Pointer on array for version information Mva (arraysize 30 Byte).

**Returns:**

None.

#### 2.1.5.11 COLORDLL\_API unsigned char Mva\_StartCalibration (unsigned char *SelectedSensor*)

Starts calibration of true color sensors.

After executing calibration data are available in **glibAdjust()** (p. 22) and **glibScale()** (p. 23).

The first function call starts **Mdi\_InitLibrary()** (p. 19), if **Mdi\_InitLibrary()** (p. 19) was not called before.

**Parameters:**

*SelectedSensor* =0 Starts calibration of all sensors.  
=1..10 Starts calibration of sensor 1..10.

**Return values:**

**glibAdjust()** (p. 22) Contains offset values, black-white offset values and calibration matrices of the sensors.

**glibScale()** (p. 23) Contains scale factors of the sensors.

**Returns:**

Error indicates the status of the calibration.  
=0 Calibration ok  
!=0 Calibration failed. See **Mdi\_InitLibrary()** (p. 19) for details.

#### 2.1.5.12 COLORDLL\_API int SetCallbackProcAddr (long *IProcAdr*, int *iIndex*)

Initialisation of the function pointer for the callback functions.

**Parameters:**

*IProcAdr* - Adress of the function to initialise.

NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

***lIndex*** - Index of the function to be initialised.

**Return values:**

***None.***

**Returns:**

ReturnValue indicates the status of the function pointer initialisation.

=1 Function pointer set successfully.

=0 Function pointer not set.

**Programming example to initialise the function pointer:**

```
// Variable for the address
static long lProcAdr;
// Temporary error code
int iErr = 0;
// Error code to return
int ReturnError = 0;

// Copy the address of the self-programmed function pCol2InitGlobals()
lProcAdr = (long>(&my_InitGlobals ));
// Function call to set the address for MDI_INITGLOBALS
if (iErr = SetCallbackProcAddr((long)lProcAdr, MDI_INITGLOBALS ) != 1)
    ReturnError = 1;
lProcAdr = (long>(&my_LoadTargetXYZ ));
if (iErr = SetCallbackProcAddr((long)lProcAdr, MVA_LOADTARGETXYZ ) != 1)
    ReturnError = 1;
lProcAdr = (long>(&my_LoadSettings ));
if (iErr = SetCallbackProcAddr((long)lProcAdr, MVA_LOADSETTINGS ) != 1)
    ReturnError = 1;
lProcAdr = (long>(&my_SaveSettings ));
if (iErr = SetCallbackProcAddr((long)lProcAdr, MVA_SAVESETTINGS ) != 1)
    ReturnError = 1;
lProcAdr = (long>(&my_SetTargetColor ));
if (iErr = SetCallbackProcAddr((long)lProcAdr, MVA_SETTARGETCOLOR ) != 1)
    ReturnError = 1;
lProcAdr = (long>(&my_StartADCReturnInt ));
if (iErr = SetCallbackProcAddr((long)lProcAdr, MDI_STARTADCRETURNINT ) != 1)
    ReturnError = 1;
lProcAdr = (long>(&my_StartADCReturnFloat ));
if (iErr = SetCallbackProcAddr((long)lProcAdr, MDI_STARTADCRETURNFLOAT ) != 1)
    ReturnError = 1;
lProcAdr = (long>(&my_MeasuringError ));
if (iErr = SetCallbackProcAddr((long)lProcAdr, MDI_MEASURINGERROR ) != 1)
    ReturnError = 1;
lProcAdr = (long>(&my_LightingOn ));
if (iErr = SetCallbackProcAddr((long)lProcAdr, MDI_LIGHTINGON ) != 1)
    ReturnError = 1;
lProcAdr = (long>(&my_LightingOff ));
if (iErr = SetCallbackProcAddr((long)lProcAdr, MDI_LIGHTINGOFF ) != 1)
    ReturnError = 1;
```

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

## 2.1.6 Variable Documentation

### 2.1.6.1 COLORDLL\_API float glibAdjust[ ]

Contains the values for the adjustment of the sensors.

total array size: 1624

array size used: 15x number of true color sensors.

Order:

nTC: Index true color sensors

m: Index channel

OFFm: Offset of channel m

HDOFFm: Black-White-Offset of channel m

MTCnTC: Correction matrix of true color sensor nTC

$glibAdjust = \{OFF1, \dots, OFFm, HDOFF1, \dots, HDOFFm, MTC1, \dots, MTCnTC\}$

Example:

- 1 true color sensor - Offset OFFX1, OFFY1, OFFZ1 - Black-White-Offset HDOFFX1, HDOFFY1, HDOFFZ1 - Matrix mtc1\_0, ..., mtc1\_8

- 1 true color sensor - OFFX2, OFFY2, OFFZ2 - HDOFFX2, HDOFFY2, HDOFFZ2 - mtc2\_0, ..., mtc2\_8

$glibAdjust[30] = \{OFFX1, OFFY1, OFFZ1, OFFX2, OFFY2, OFFZ2, HDOFFX1, HDOFFY1, HDOFFZ1, HDOFFX2, HDOFFY2, HDOFFZ2, mtc1_0, \dots, mtc1_8, mtc2_0, \dots, mtc2_8\}$

Definition at line 725 of file JenColorDLL.h.

### 2.1.6.2 COLORDLL\_API unsigned int glibMeasure[ ]

Contains the number of measurements each sensor for averaging.

Range of values: 1..65535

total array size: 17

array size used: Number of true color sensors.

Order:

n: Index sensors

Mn: Number of measurements of sensor n for averaging

$glibMeasure = \{M1, \dots, Mn\}$

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

Example:

- 1 true color sensor - 99 measurements
  - 1 true color sensor - 2048 measurements
  - 1 true color sensor - 801 measurements
- `glibMeasure[3] = {99, 2048, 801}`

Definition at line 637 of file JenColorDLL.h.

### 2.1.6.3 COLORDLL\_API float glibResult[ ]

Result array of the library JenColorDLL. Contains the sensor results after a measurement.

total array size: 232

array size used: 3x number of true color sensors.

Order:

- nTC: Index true color sensors
  - EnTC: (3,1) Result matrix of true color sensor nTC
- `glibResult = {E1TC, ..., EnTC}`

Example:

- 1 true color sensor - Result X1, Y1, Z1
  - 1 true color sensor - Result X2, Y2, Z2
- `glibResult[6] = {X1, Y1, Z1, X2, Y2, Z2}`

Definition at line 746 of file JenColorDLL.h.

### 2.1.6.4 COLORDLL\_API signed char glibScale[ ]

Contains the scalefactor of the sensors, calculated during calibration.

Range of values: -127..+127

total array size: 17

array size used: Number of true color sensors.

Order:

- n: Index sensors
  - SFn: Scalefactor of sensor n
- `glibScale = {SF1, ..., SFn }`

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

Example:

- 1 true color sensor - Scalefactor  $2^{-1}$
  - 1 true color sensor - Scalefactor  $2^3$
  - 1 true color sensor - Scalefactor  $2^0$
- `glibScale[3] = {-1, 3, 0}`

Definition at line 702 of file JenColorDLL.h.

### 2.1.6.5 COLORDLL\_API unsigned char glibSensor[]

Contains information about the sensor like the measure sequence, resolution and lighting.

1. Sequence of the function calls during adjustment of a measurement (bit-coded):

- 0b0000 0001 = Offset adjustment
  - 0b0000 0010 = Scaling
  - 0b0000 0100 = Black-White-offset adjustment
  - 0b0000 1000 = linear measurement adjustment
- Range of values: 0..15

Example:

- 0b0000 0000 = no adjustment
- 0b0000 1101 = adjustment of a measurement without scaling

2. Resolution of a sensor in bit:

Range of values: 8..16

3. Coded system lighting of the sensors:

- 0 = System without lighting / Measurement without lighting
- 1 = Measurement with lighting
- 2 = Measurement with backlight compensation

Range of values: 0..2

total array size: 51

array size used: 3x number of true color sensors.

Order:

- n: Index sensors
- MSn: measure sequence of sensor n

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

BITn: Resolution of sensor n in bit

SBn: System lighting of sensor n

$glibSensor = \{MS1, \dots, MSn, BIT1, \dots, BITn, SB1, \dots, SBn\}$

Example:

- 1 true color sensor - Read out the ADC value - 10-bit - System without lighting
- 1 true color sensor - Complete measure sequence - 12-bit - Measurement with backlight compensation
- 1 true color sensor - without Black-White-offset adjustment - 16-bit - Measurement with lighting

$glibSensor[9] = \{0, 15, 7, 10, 12, 16, 0, 2, 1\}$

Definition at line 679 of file JenColorDLL.h.

#### **2.1.6.6 TMDi\_InitGlobals PMdi\_InitGlobals**

Variable for imported function (internal use).

#### **2.1.6.7 TMDi\_LightingOff PMdi\_LightingOff**

Variable for imported function (internal use).

#### **2.1.6.8 TMDi\_LightingOn PMdi\_LightingOn**

Variable for imported function (internal use).

#### **2.1.6.9 TMDi\_MeasuringError PMdi\_MeasuringError**

Variable for imported function (internal use).

#### **2.1.6.10 TMDi\_StartADCReturnFloat PMdi\_StartADCReturnFloat**

Variable for imported function (internal use).

#### **2.1.6.11 TMDi\_StartADCReturnInt PMdi\_StartADCReturnInt**

Variable for imported function (internal use).

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

#### **2.1.6.12 TMva\_LoadSettings PMva\_LoadSettings**

Variable for imported function (internal use).

#### **2.1.6.13 TMva\_LoadTargetXYZ PMva\_LoadTargetXYZ**

Variable for imported function (internal use).

#### **2.1.6.14 TMva\_SaveSettings PMva\_SaveSettings**

Variable for imported function (internal use).

#### **2.1.6.15 TMva\_SetTargetColor PMva\_SetTargetColor**

Variable for imported function (internal use).

VERSION AMENDMENTS		
NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

## 3 JENCOLORLibrary Page Documentation

### 3.1 Todo List

**Global TMdi\_InitGlobals (p. 9)** Change the status value for function call during measurement, number of true color sensors and calibration targets to your applications need.

**Global TMdi\_LightingOff (p. 9)** If your system uses lighting, write a routine to turn off the light.

**Global TMdi\_LightingOn (p. 10)** If your system uses lighting, write a routine to turn on the light.

**Global TMdi\_MeasuringError (p. 10)** Write your own error handling routine if a sensor causes a zero signal or signal overflow.

**Global TMdi\_StartADCReturnFloat (p. 11)** Write your own AD conversation or interface functions to read out a true color sensor and return float values. With this function the internal averaging cannot be used.

**Global TMdi\_StartADCReturnInt (p. 12)** Write your own AD conversation or interface functions to read out a true color sensor and return integer values. With this function the internal averaging can be used.

**Global TMva\_LoadSettings (p. 13)** Reload the arrays glibAdjust, glibMeasure, glibScale and glibSensor for each sensor from a non-volatile storage. (Saved before in **TMva\_SaveSettings()** (p. 14).)

**Global TMva\_LoadTargetXYZ (p. 13)** The XYZ target values have to be adjusted on the application! Measure your calibration targets with a calibrated spectrometer and insert the XYZ target values in this function.

**Global TMva\_SaveSettings (p. 14)** Save the arrays glibAdjust, glibMeasure, glibScale and glibSensor for each sensor on a non-volatile storage.

**Global TMva\_SetTargetColor (p. 15)** Adjust this function to your application.

i.e. this function can be automated if a multicolor LED or a display is measured. The LED/display can be set automatically to the specific color without asking the user to do this.

## Index

COLORDLL\_API  
    JenColorDLL.h, 8

glibAdjust  
    JenColorDLL.h, 22

glibMeasure  
    JenColorDLL.h, 22

glibResult  
    JenColorDLL.h, 23

glibScale  
    JenColorDLL.h, 23

glibSensor  
    JenColorDLL.h, 24

JenColorDLL.h, 4  
    COLORDLL\_API, 8  
    glibAdjust, 22  
    glibMeasure, 22  
    glibResult, 23  
    glibScale, 23  
    glibSensor, 24  
    Mcm\_deltaEab, 16  
    Mcm\_GetVersion, 16  
    Mcm\_RGBtoXYZ, 17  
    Mcm\_XYZtoLab, 17  
    Mcm\_XYZtoLuv, 17  
    Mcm\_XYZtoRGB, 18  
    Mcm\_XYZtoxyY, 18  
    MDI\_INITGLOBALS, 16  
    Mdi\_InitLibrary, 18  
    MDI\_LIGHTINGOFF, 16  
    MDI\_LIGHTINGON, 16  
    MDI\_MEASURINGERROR, 16  
    MDI\_STARTADCRETURNFLOAT,  
        16  
    MDI\_STARTADCRETURNINT, 16  
    Mva\_GetAdjustedValues, 19  
    Mva\_GetVersion, 19  
    MVA\_LOADSETTINGS, 16  
    MVA\_LOADTARGETXYZ, 16  
    MVA\_SAVESETTINGS, 16  
    MVA\_SETTARGETCOLOR, 16  
    Mva\_StartCalibration, 20  
    PMdi\_InitGlobals, 25  
    PMdi\_LightingOff, 25  
    PMdi\_LightingOn, 25  
    PMdi\_MeasuringError, 25  
    PMdi\_StartADCReturnFloat, 25  
    PMdi\_StartADCReturnInt, 25  
    PMva\_LoadSettings, 25  
    PMva\_LoadTargetXYZ, 26  
    PMva\_SaveSettings, 26  
    PMva\_SetTargetColor, 26  
    PROGS, 16  
    SetCallbackProcAddr, 20  
    TMdi\_InitGlobals, 9  
    TMdi\_LightingOff, 9  
    TMdi\_LightingOn, 10  
    TMdi\_MeasuringError, 10  
    TMdi\_StartADCReturnFloat, 11  
    TMdi\_StartADCReturnInt, 12  
    TMva\_LoadSettings, 12  
    TMva\_LoadTargetXYZ, 13  
    TMva\_SaveSettings, 14  
    TMva\_SetTargetColor, 15

Mcm\_deltaEab  
    JenColorDLL.h, 16

Mcm\_GetVersion  
    JenColorDLL.h, 16

Mcm\_RGBtoXYZ  
    JenColorDLL.h, 17

Mcm\_XYZtoLab  
    JenColorDLL.h, 17

Mcm\_XYZtoLuv  
    JenColorDLL.h, 17

Mcm\_XYZtoRGB  
    JenColorDLL.h, 18

Mcm\_XYZtoxyY  
    JenColorDLL.h, 18

Software Description JENCOLOR Library  
JenColorDLL

VERSION AMENDMENTS

NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

MDI_INITGLOBALS JenColorDLL.h, 16	PMva_LoadTargetXYZ JenColorDLL.h, 26
Mdi_InitLibrary JenColorDLL.h, 18	PMva_SaveSettings JenColorDLL.h, 26
MDI_LIGHTINGOFF JenColorDLL.h, 16	PMva_SetTargetColor JenColorDLL.h, 26
MDI_LIGHTINGON JenColorDLL.h, 16	PROGS JenColorDLL.h, 16
MDI_MEASURINGERROR JenColorDLL.h, 16	SetCallbackProcAddr JenColorDLL.h, 20
MDI_STARTADCRETURNFLOAT JenColorDLL.h, 16	TMdi_InitGlobals JenColorDLL.h, 9
MDI_STARTADCRETURNTINT JenColorDLL.h, 16	TMdi_LightingOff JenColorDLL.h, 9
Mva_GetAdjustedValues JenColorDLL.h, 19	TMdi_LightingOn JenColorDLL.h, 10
Mva_GetVersion JenColorDLL.h, 19	TMdi_MeasuringError JenColorDLL.h, 10
MVA_LOADSETTINGS JenColorDLL.h, 16	TMdi_StartADCReturnFloat JenColorDLL.h, 11
MVA_LOADTARGETXYZ JenColorDLL.h, 16	TMdi_StartADCReturnInt JenColorDLL.h, 12
MVA_SAVESETTINGS JenColorDLL.h, 16	TMva_LoadSettings JenColorDLL.h, 12
MVA_SETTARGETCOLOR JenColorDLL.h, 16	TMva_LoadTargetXYZ JenColorDLL.h, 13
Mva_StartCalibration JenColorDLL.h, 20	TMva_SaveSettings JenColorDLL.h, 14
PMdi_InitGlobals JenColorDLL.h, 25	TMva_SetTargetColor JenColorDLL.h, 15
PMdi_LightingOff JenColorDLL.h, 25	
PMdi_LightingOn JenColorDLL.h, 25	
PMdi_MeasuringError JenColorDLL.h, 25	
PMdi_StartADCReturnFloat JenColorDLL.h, 25	
PMdi_StartADCReturnInt JenColorDLL.h, 25	
PMva_LoadSettings JenColorDLL.h, 25	

NO.	VERSION	APPROVED
1	V 1.0	29.07.2008

---

For further information please contact:

**MAZeT GmbH**

**Sales Department:**

Göschwitzer Strasse 32

07745 Jena, Germany

Tel: +49 3641 2809-0

Fax: +49 3641 2809-12

Email: [sales@MAZeT.de](mailto:sales@MAZeT.de)

Website: <http://www.MAZeT.de>