

Chroma Meter CS-200 DLL Reference



Chroma Meter CS-200

DLL Reference

Chroma Meter CS-200 DLL Reference

Notes concerning software and associated documentation:

- When Chroma Meter CS-200 is used connected with a PC or other devices, USB driver for CS-200 is needed.
- USB driver for CS-200 and/or accompanying manuals (hereinafter referred to as "SOFTWARE") is copyrighted by Konica Minolta.
- Alteration of this software without the express permission of Konica Minolta is strictly prohibited. Unauthorized copying of this software, in whole or in part, is strictly prohibited.
- This software can be used by customers who purchased Chroma Meter CS-200 and agree with the terms of the End User License Agreement.
- The commands and other information described in these documents cannot be used for purposes related to products for commercial sales without the express permission of Konica Minolta, Inc.
- This document is designed for users who are familiar with the fundamentals of PC communications.
- Reproduction of this document, its contents, or any portion thereof without the permission of Konica Minolta, Inc. is strictly prohibited.
- Contents of this document are subject to change without notice.
- Although every effort has been taken to ensure the accuracy of the information in this document, if there are any questions or if any errors are found, please feel free to contact the nearest Konica Minolta representative.
- Konica Minolta, Inc. is not responsible for any consequences resulting from the information herein.
- Company names and product names that appear within this document are trademarks or registered trademarks of their respective companies.

2005.01.26

KONICA MINOLTA, INC.

Chroma Meter CS-200 DLL Reference

Index

1. Preparations	4
2. Using the DLL.....	7
2.1 Installing the DLL	7
2.2 Functions.....	7
2.2.1 int_usb	7
2.2.2 end_usb.....	8
2.2.3 get_num.....	8
2.2.4 write64_usb	8
2.2.5 read64_usb.....	8
3. Sample Code	9
3.1 Visual C++	9
3.1.1 Linking with DLL	9
3.1.2 Sample Code.....	9
3.2 Visual Basic	9
3.2.1 Linking with DLL	12
3.2.2 Sample Code.....	12

Chroma Meter CS-200 DLL Reference

1. Preparations

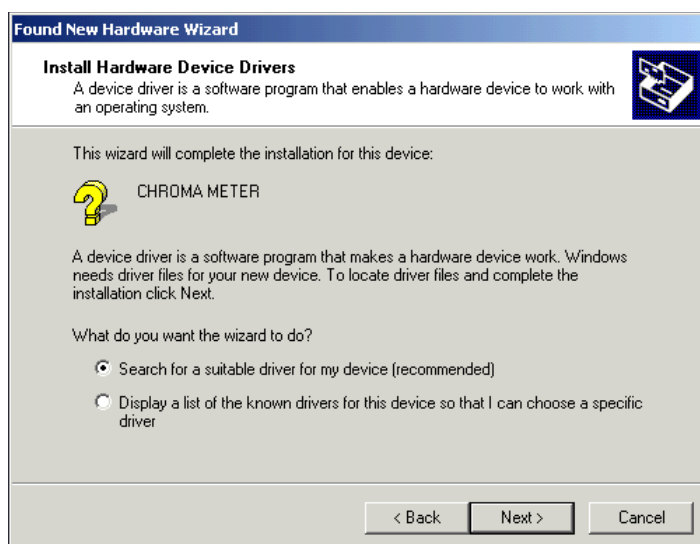
1.1 Installing the USB Driver files

When using the CS-200 for USB communication, the USB Driver files must be installed.

- (1) Connect the CS-200 to the computer using the USB connecting cable and switch on the power of the CS-200. The dialog box shown below will appear. Click Next.



- (2) The dialog box shown below will appear. Select "Search for a suitable driver for my device". Click Next.



Chroma Meter CS-200 DLL Reference

(3) The dialog box shown below will appear. Check “Specify a location” and click Next.



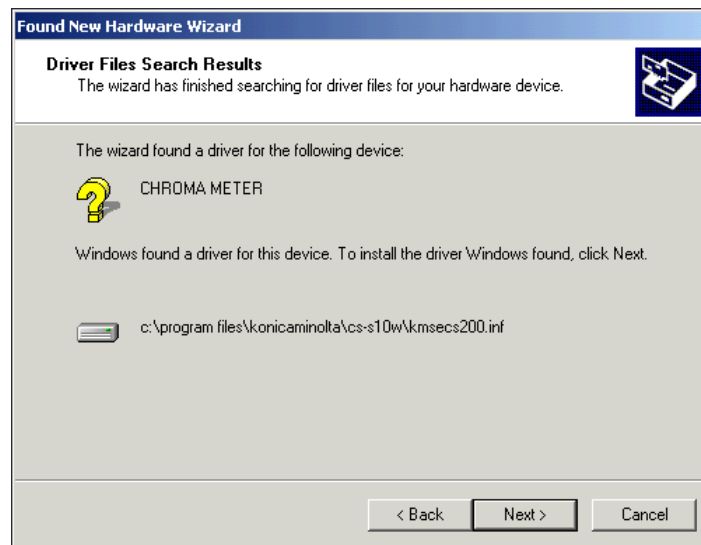
(4) The dialog box shown below will appear. Please select the following file path.

- When CS-S10w is installed, select the install path.
- When CS-S10w is not installed, select the folder in which the USB driver file was saved when it was downloaded from the web site.



Chroma Meter CS-200 DLL Reference

(5) The dialog box shown below will appear. Click Next.



(6) The dialog box shown below will appear. Click Finish.



Chroma Meter CS-200 DLL Reference

2. Using the DLL

* Applications will run in 64-bit environments, but a 64-bit version of applications cannot be created.

2.1 Installing the DLL

Copy the file Kmsecs200.dll into one of the following locations:

(The latest version of the USB driver, including Kmsecs200.dll, can be downloaded from the Konica Minolta, Inc. web site.)

- The directory containing the executable (*.exe) file
- The current directory
 - The current directory for Visual Basic is the directory containing VB32.EXE.
- The Windows system directory (normally, \Windows\System32)

After making the executable (*.exe) file, install the DLL file above into the directory containing the executable (*.exe) file.

2.2 Functions

int_usb	Opens the specified instrument.
end_usb	Closes the specified instrument.
get_num	Requests the number of the connected instrument.
write64_usb	Sends data or commands to the specified instrument. (DMA)
read64_usb	Receives response or data from the specified instrument. (DMA)

2.2.1 int_usb

Command	int int_usb(int Index)	
Return	0	Function completed normally.
	-1	The Index value is incorrect.
	-2	The instrument is not connected.
	-3	Failed to open the send pipe. (FIFO)
	-4	Failed to open the receive pipe. (FIFO)
	-5	Failed to open the send pipe. (DMA)
	-6	Failed to open the receive pipe. (DMA)
Parameter	Index	Instrument ID(0 to 126)
Explanation	Opens the pipe to the instrument specified by the index. When open fails, closes all pipes. When disconnecting the instrument, call "end_usb". [Comment] Instrument ID is a number from 0 to 126. When connecting only one instrument, instrument ID is 0. When connecting two or more instruments, instrument ids are allocated in order from 0.	

Chroma Meter CS-200 DLL Reference

2.2.2 end_usb

Command	int end_usb(int Index)	
Return	0	Function completed normally
	-1	The Index value is incorrect.
Parameter	Index	Instrument ID(0 to 126)
Explanation	Closes the pipe to the instrument specified by the index.	

2.2.3 get_num

Command	int get_num()	
Return	Connected instrument count	
Parameter		
Explanation	Gets the count of the connected instruments.	

2.2.4 write64_usb

Command	int write64_usb(int Index, char* Cmd, int Timeout, int WriteLen)	
Return	0	Function completed normally.
	-1	The Index value is incorrect.
	-2	Pipe not opened.
	-3	Send error.
Parameters	Index	Instrument ID(0 to 126)
	Cmd	Send string (maximum length: 64)
	Timeout	Unused data(Input value greater than 1.)
	WriteLen	Send string length
Explanation	Sends string to the instrument specified by index.	

2.2.5 read64_usb

Command	int read64_usb(int Index, char* Dat, int Timeout, int ReadLen)	
Return	0	Function completed normally.
	-1	The Index value is incorrect.
	-2	Pipe not opened.
	-3	Receive error.
Parameters	Index	Instrument ID(0 to 126)
	Dat	Receive string
	Timeout	Unused data(Input value greater than 1.)
	ReadLen	Receive string length (Always input 250.)
Explanation	Receives string from the instrument specified by index.	

Chroma Meter CS-200 DLL Reference

3. Sample Code

3.1 Visual C++

3.1.1 Linking with DLL

- a. (1) Linking with DLL, and getting the handle.

Get the DLL handle. Using this handle, perform communication.

```
HINSTANCE m_hDll = LoadLibrary("kmsecs200.dll");
```

- b. (2) Get the function address.

- Define function pointers.

```
typedef INT (CALLBACK* USB_INI)(INT);
```

```
typedef INT (CALLBACK* USB_NUM)(VOID);
```

```
typedef INT (CALLBACK* USB_IO)(INT, LPSTR, INT, INT);
```

- Get the address corresponding to each function. Using this address, create your own software.

```
USB_INI int_usb = (USB_INI)GetProcAddress(m_hDll, "int_usb");
```

```
USB_INI end_usb = (USB_INI)GetProcAddress(m_hDll, "end_usb");
```

```
USB_NUM get_num = (USB_NUM)GetProcAddress(m_hDll, "get_num");
```

```
USB_IO write64_usb = (USB_IO)GetProcAddress(m_hDll, "write64_usb");
```

```
USB_IO read64_usb = (USB_IO)GetProcAddress(m_hDll, "read64_usb");
```

- c. (3) Release the linked DLL.

```
FreeLibrary(m_hDll);
```

3.1.2 Sample Code

```
// Define type
typedef INT (CALLBACK* USB_INI)(INT);
typedef INT (CALLBACK* USB_NUM)(VOID);
typedef INT (CALLBACK* USB_IO)(INT, LPSTR, INT, INT);

// [Examples]Case measure

// Link the DLL
HINSTANCE hDll = LoadLibrary("kmsecs200.dll");
```

Chroma Meter CS-200 DLL Reference

```
// Get the function address
USB_INI int_usb = (USB_INI)GetProcAddress(hDll, "int_usb");
USB_INI end_usb = (USB_INI)GetProcAddress(hDll, "end_usb");
USB_IO write64_usb = (USB_IO)GetProcAddress(hDll, "write64_usb");
USB_IO read64_usb = (USB_IO)GetProcAddress(hDll, "read64_usb");

// Open the pipe
int_usb(0);

char cBuf[250];

// Set remote mode
char cRemote[]={"RMT,1\r\n"};
write64_usb(0, cRemote, 1, 7);
read64_usb(0, cBuf, 1, 250);

// Start measurement
char cMes[]={"MES,1\r\n"};
write64_usb(0, cMes, 1, 7);
read64_usb(0, cBuf, 1, 250);

// Get measuring time
CString str(cBuf);
int nTime = atoi(str.Mid(5, 2));

// Create wait time of (measuring time - 0.5)s
DWORD dwTime = (DWORD)(nTime*1000 - 500);
Sleep(dwTime);
```

Chroma Meter CS-200 DLL Reference

```
// Receive measurement data
char cMdr[]={"MDR,0\r\n"};
while(1){
    write64_usb(0, cMdr, 1, 7);
    read64_usb(0, cBuf, 1, 250);

    CString str(cBuf);
    if (str.Left(4) == _T("ER02")){
        // measuring
        // Wait 0.3s before resending command
        Sleep(300);
    }else{
        // OK or ER
        break;
    }
}

// Case OK, data are stored in cBuf

// Close the pipe
end_usb(0);

// Release the linked DLL
FreeLibrary(hDll);
```

Chroma Meter CS-200 DLL Reference

3.2 Visual Basic

3.2.1 Linking with DLL

- d. Define the DLL functions. Using defined functions, perform communication.

```
Public Declare Function int_usb Lib "kmsecs200.dll" (ByVal Index As Long) As Long
Public Declare Function end_usb Lib "kmsecs200.dll" (ByVal Index As Long) As Long
Public Declare Function get_num Lib "kmsecs200.dll" () As Long
Public Declare Function write64_usb Lib "kmsecs200.dll" (ByVal Index As Long, ByVal
    Cmd As String, ByVal TimeOut As Long, ByVal WriteLen As Long) As Long
Public Declare Function read64_usb Lib "kmsecs200.dll" (ByVal Index As Long, ByVal Dat
    As String, ByVal TimeOut As Long, ByVal ReadLen As Long) As Long
```

3.2.2 Sample Code

```
' Define functions
Public Declare Function int_usb Lib "kmsecs200.dll" (ByVal Index As Long) As Long
Public Declare Function end_usb Lib "kmsecs200.dll" (ByVal Index As Long) As Long
Public Declare Function write64_usb Lib "kmsecs200.dll" (ByVal Index As Long, ByVal Cmd
    As String, ByVal TimeOut As Long, ByVal WriteLen As Long) As Long
Public Declare Function read64_usb Lib "kmsecs200.dll" (ByVal Index As Long, ByVal Dat As
    String, ByVal TimeOut As Long, ByVal ReadLen As Long) As Long

' [Examples]Case measure

' Open the pipe
int_usb 0

' Define parameter
Dim strSend As String
Dim strRecv As String
Dim sngTime As Single
```

Chroma Meter CS-200 DLL Reference

```
' Set remote mode
strSend = "RMT,1" + vbCr + vbLf
strRecv = String(250, Chr(0))
write64_usb 0, strSend, 1, Len(strSend)
read64_usb 0, strRecv, 1, 250

' Start measurement
strSend = "MES,1" + vbCr + vbLf
strRecv = String(250, Chr(0))
write64_usb 0, strSend, 1, Len(strSend)
read64_usb 0, strRecv, 1, 250

If Left(rBuf, 2) = "OK" Then
    ' Get measuring time
    sngTime = CSng(Mid(rBuf, 6, 2))
    ' Create wait time of (measuring time - 0.5)s
    WaitSec(sngTime - 0.5)
Else
    ' Error
End If

' Receive measurement data
strSend = "MDR,0" + vbCr + vbLf
Do
    strRecv = String(250, Chr(0))
    write64_usb 0, strSend, 1, Len(strSend)
    read64_usb 0, strRecv, 1, 250
    If Left(strRecv,4) = "ER02" Then
        ' Wait 0.3s before resending command
        WaitSec(0.3)
    Else
        Exit Do
    End If
Loop While(1)
```

Chroma Meter CS-200 DLL Reference

```
' Case OK, Data are stored in strRecv
```

```
' Close the pipe
```

```
end_usb 0
```

```
' Wait function example
```

```
Sub WaitSec(t As Single)
```

```
    Dim start_t, end_t As Single
```

```
    Dim passed_t As Single
```

```
    start_t = Timer
```

```
    Do
```

```
        end_t = Timer
```

```
        passed_t = end_t - start_t
```

```
        If passed_t < 0 Then passed_t = passed_t + 24# * 60# * 60#
```

```
        DoEvents
```

```
    Loop Until passed_t >= t
```

```
End Sub
```