



---

# **Programmer's Guide**

## **Scanning Monochromator**

### **MonoScan-2000**

Revision 102006



© Copyright 2005  
Mikropack GmbH  
Maybachstrasse 11  
D-73760 Ostfildern  
Germany  
Tel.: +49 – (0)711 – 34 16 96 0  
Fax: +49 – (0)711 – 34 16 96 85  
E-Mail: [Info@mikropack.de](mailto:Info@mikropack.de)



---

## MonoScan-2000 Interface-DLL

The following functions are exported by MonoScandrv.dll:

### **MONO\_OpenConnection**

```
long __stdcall MONO_OpenConnection(char* PortName);
```

Returns the handle value for the specified Comport as long integer.

This value has to be stored locally for all other functions.

Returns a negative value in case of error.

Portname has to be e.g. "Com1", "Com2"

### **MONO\_CloseConnection**

```
short __stdcall MONO_CloseConnection(long Handle);
```

Close the connection opened before.

Returns 1 on success else -1

long Handle: Handle of the comport returned by MONO\_OpenConnection

### **MONO\_GetDeviceData**

```
short __stdcall MONO_GetDeviceData(long Handle, unsigned short Node, long* CalibrationData,  
                                   long SerialNo);
```

Get the calibration data and the serialnumber of the device

long Handle: The handle to the comport returned by MONO\_OpenConnection

unsigned short Node: has to be 0 (zero)

long\* CalibrationData: Array of long integers, has to contain 92 elements

long SerialNo: The serialnumber of the Device, at least 10 characters



---

### **MONO\_SetWavelength**

```
short __stdcall MONO_SetWavelength(long Handle, unsigned short Node,  
                                   double WavelengthInNm, long* CalibrationData);
```

Set the MonoScan-2000 to a specified wavelength in nm.

Returns 1 on success else a negative number.

Long Handle: Handle of the comport returned by MONO\_OpenConnection

unsigned short Node: must be 0 (zero)

short WavelengthInNm: desired wavelength in nm

long\* CalibrationData: Array filled by MONO\_GetDeviceData

### **MONO\_GetWavelength**

```
double __stdcall MONO_GetWavelength(long Handle, unsigned short Node,  
                                    long* CalibrationData);
```

returns the actual position of the MonoScan-2000 in nm.

Long Handle: Handle of the comport returned by MONO\_OpenConnection

unsigned short Node: must be 0 (zero)

long\* CalibrationData: Array filled by MONO\_GetDeviceData

### **MONO\_RunReferenceSequence**

```
short __stdcall MONO_RunReferenceSequence(long Handle,unsigned short Node);
```

MonoScan will run to its internal reference point and reset the internal position

Long Handle: Handle of the comport returned by MONO\_OpenConnection

unsigned short Node: must be 0 (zero)

### **MONO\_HomingRunActive**

```
short __stdcall MONO_HomingRunActive(long Handle,unsigned Node);
```

call this function sequentially to see if reference sequence has completed

returns 1 if reference sequence is still active else 0.

Long Handle: Handle of the comport returned by MONO\_OpenConnection

unsigned short Node: must be 0 (zero)



---

### **MONO\_PositionReached**

short \_\_stdcall MONO\_PositionReached(long Handle,unsigned short Node);

call this function to see if the desired position has been reached.

Returns 0 if still moving else 1

Long Handle: Handle of the comport returned by MONO\_OpenConnection

unsigned short Node: must be 0 (zero)

### **MONO\_SetPosition**

short \_\_stdcall MONO\_SetPosition(long Handle,unsigned short Node,long Position);

call this function to set an absolute user defined position. Wavelength is not linear to position

Returns 0 in case of error else 1

long Handle: Handle of the comport returned by MONO\_OpenConnection

unsigned short Node: must be 0 (zero)

long position: absolute position value

### **MONO\_GetPosition**

long \_\_stdcall MONO\_GetPosition(long Handle,unsigned short Node);

call this function to get the absolute position in steps (arbitrary unit)

long Handle: handle of the comport returned by MONO\_OpenConnection

unsigned short Node: must be 0 (zero)



---

## Direct Serial Communication (ASCII Comands)

If you do not want to use the DLL-functions you may directly communicate with the MonoScan2000 using ASCII commands.

The following steps are required:

1. Establish serial connection from PC to MonoScan-2000
2. Read device information (Calibrationdata and Serialnumber)  
(not necessarily needed)
3. Set an absolute position
4. repeat step 3 as long as you need
5. Close serial connection

### ASCII Commands

<i>Command</i>	<i>Description</i>	<i>Example</i>
GOHOSEQ	Reference run is executed. Runs to the lower endpoint (approx 200nm) and sets this position to 0 (zero)	GOHOSEQ [CR]
LA<Pos>	Sets absolute position in steps Command M starts positioning	LA-5000 [CR] M [CR]
LR<Pos>	Sets relative position in steps Command M starts positioning	LR100 [CR] M [CR]
POS	Returns actual position in steps	POS [CR]
HO	Sets current position to 0 (zero). A reference run executed before will be lost	HO [CR]
GPROGSEQ	Returns the data stored in the MonoScan-2000	GPROGSEQ [CR]

All Commands have to be completed by Carriage return [CR]



---

## Stored Data

<i>Stored data</i>	<i>Comment</i>
JMP1	Internal
lr11220001	SerialNumber
lr-1234	Absolute position for 200nm (not valid)
lr-1734	
lr-2497	
lr-3087	
lr-3665	
lr-4201	Absolute position for 250nm
lr-4750	
lr-5306	
...	
lr-46911	Absolute Position for 1100nm (not valid)
A1	
LR5000	Internal
M	Internal
delay20	Internal
gohoseq	Start reference run

Everytime the MonoScan-2000 is powered on a reference run is performed.



---

## End User License Agreement for Software (EULA)

This End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and Mikropack GmbH ("Mikropack") for the Mikropack MonoScandrv software, which includes computer software and may include associated media, printed materials, and "online" or electronic documentation ("SOFTWARE"). By installing, copying, or otherwise using the SOFTWARE, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, do not install, copy or use the SOFTWARE.

The SOFTWARE is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE is licensed, not sold.

### 1. License to Use

Mikropack grants you a non-exclusive and non-transferable license for the internal use only of the SOFTWARE and any error corrections provided by Mikropack.

### 2. Restrictions

SOFTWARE is confidential and copyrighted. Title to SOFTWARE and all associated intellectual property rights is retained by Mikropack and/or its licensors. You may not make copies of SOFTWARE, other than a single copy of SOFTWARE for archival purposes. Unless enforcement is prohibited by applicable law, you may not modify, decompile, or reverse engineer SOFTWARE. No right, title or interest in or to any trademark, service mark, logo or trade name of Mikropack or its licensors is granted under this EULA.

### 3. Limited Warranty

Mikropack warrants to you that for a period of six (6) months from the date of shipping, as evidenced by a copy of the receipt, the media on which SOFTWARE is furnished (if any) will be free of defects in materials and workmanship under normal use. Except for the foregoing, SOFTWARE is provided "AS IS". Your exclusive remedy and Mikropack's entire liability under this limited warranty will be at Mikropack's option to replace SOFTWARE media or refund the fee paid for SOFTWARE.

### 4. Disclaimer of Warranty

Unless specified in this EULA, all express or implied conditions, representations and warranties, including any implied warranty of merchantability, fitness for a particular purpose or non-infringement are disclaimed, except to the extent that these disclaimer are held to be legally invalid.

### 5. Limitation of Liability

To the extent not prohibited by law, in no event will Mikropack or its licensors be liable for any lost revenue, profit or data, or for special, indirect, consequential, incidental or punitive damages, however caused regardless of the theory of liability, arising out of or related to the use of or inability to use SOFTWARE, even if Mikropack has been advised of the possibility of such damages. In no event will Mikropack's liability to you, whether in contract, tort (including negligence), or otherwise, exceed the amount paid by you for Software under this EULA. The foregoing limitations will apply even if the above stated warranty fails of its essential purpose.