

## Bode 100 - Application Note

# Automated Measurements using Multiple Bode 100 Devices



By Florian Hämmerle

© 2020 by OMICRON Lab – V1.0

Visit [www.omicron-lab.com](http://www.omicron-lab.com) for more information.

Contact [support@omicron-lab.com](mailto:support@omicron-lab.com) for technical support.

## Table of Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>The Bode Automation Interface (AI)</b>	<b>3</b>
2.1	Installation	4
2.2	How to Access the AI	4
<b>3</b>	<b>Measurement Example</b>	<b>5</b>
3.1	Initialize the Automation Interface & Connect to Bode 100	5
3.2	Configure / Create a Measurement	5
3.3	Execute the Measurement	6
3.4	Read the Results	6
3.5	Disconnect from the Device	6
<b>4</b>	<b>Using Multiple Bode 100 Devices in Parallel</b>	<b>6</b>
4.1	Hardware Considerations	6
4.1.1	Measurement Setup	6
4.1.2	USB Ports	8
4.1.3	System Requirements	9
4.2	Software Architecture	10
4.2.1	Single-Process Architecture	10
4.2.2	Multi-Process Architecture	11
<b>5</b>	<b>Summary</b>	<b>12</b>

**Note:** This document is based on .NET (C#). All findings and data shown in this document have been derived using .NET. OMICRON Lab cannot guarantee that other programming languages and compilers will behave in exactly the same way.

**Note:** All measurements in this application note have been performed with the Bode Analyzer Suite or Bode Automation Interface V3.23 and Bode 100 Revision 2. Use this version or a higher version to perform the measurements shown in this document. You can download the latest Bode Analyzer Suite version at [www.omicron-lab.com/bode-100/downloads](http://www.omicron-lab.com/bode-100/downloads)

## 1 Introduction

Bode 100 is the perfect choice to perform automated frequency-sweep measurements that are suitable for quality assurance of e.g. Piezo component manufacturing, RFID/NFC tag manufacturing or magnetics production and more.

This application note describes the basic principle of automated Bode 100 measurements and puts a focus on what needs to be considered when using multiple Bode 100 devices to perform automated tests/measurements.

**Note:** This document has been created using the **Bode 100 Revision 2** hardware and **Bode Analyzer Suite 3.23** software.

## 2 The Bode Automation Interface (AI)

The Bode 100 device can be automated on Microsoft Windows computers via the Bode Automation Interface (AI). The AI is a software interface that allows you to access all measurement possibilities of the Bode Analyzer Suite via a COM<sup>1</sup> interface or natively via .NET programming languages. The following figure shows the architecture how your software application can control the Bode 100 device that is attached to the computer via a USB 2.0 interface.

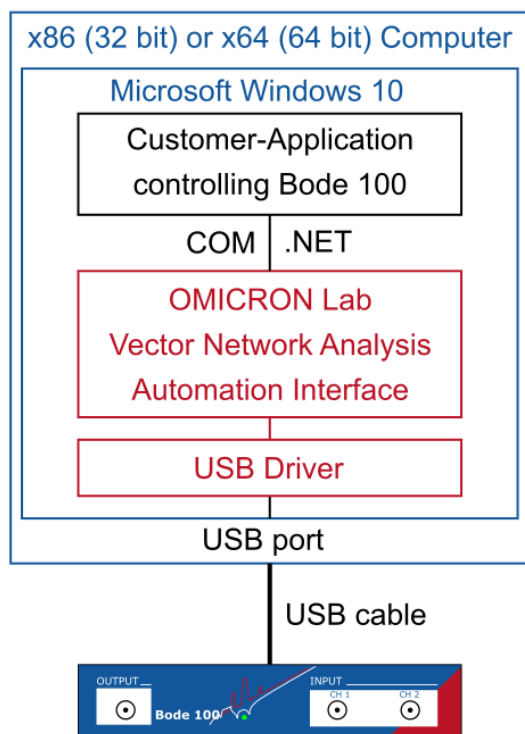


Figure 1: Automation architecture

<sup>1</sup> Component Object Model (inter process communication standard)

## 2.1 Installation

Automating the Bode 100 requires a 32-bit or 64-bit computer with Microsoft Windows 10. On the computer, the Bode Analyzer Suite respectively the Bode Automation Interface must be installed. The latest installer can be found at: <https://www.omicron-lab.com/downloads/vector-network-analysis/bode-100/>

The installer provides the USB device driver and delivers the DLLs. Furthermore, it registers the COM interface on the computer.

If you want to deliver your own software package with your own installer, we recommend that you include our Automation Interface MSI<sup>2</sup>.

## 2.2 How to Access the AI

**Note:** Independent of the method you use to access the AI, you must run the Automation Interface installer or Bode Analyzer Suite installer on your computer once. This will install the *required USB drivers* on your computer!

### Access via COM

You can access the Bode AI from any Windows programming language that supports COM. Check out the chapter “*Add AI Reference*” in the Automation Interface Reference/Help, to learn how to connect your software project to the Bode AI:

<https://www.omicron-lab.com/BodeAutomationInterfaceHelp>

### Native .NET access

If you plan to access the Bode AI from .NET programming languages, you can also use the Automation Interface NuGet package: “*OmicronLab.VectorNetworkAnalysis.AutomationInterface*”.

---

<sup>2</sup> Microsoft Installer or Windows Installer package

## 3 Measurement Example

The following simplified example uses C# and native .NET access in Visual Studio 2010 or newer. For other short examples, please check out the online documentation at:

<https://www.omicron-lab.com/BodeAutomationInterfaceHelp>.

**Note:** This is a simple basic example without any additional error handling.

### 3.1 Initialize the Automation Interface & Connect to Bode 100

After installing the Automation Interface (USB driver), add the following NuGet package to your project using the Package Manager Console:

```
PM> Install-Package OmicronLab.VectorNetworkAnalysis.AutomationInterface
```

Then create a new Automation Object and connect to Bode devices:

```
//Create new BodeAutomationInterface  
BodeAutomationInterface bodeAutomationInterface = new BodeAutomation();
```

Using the *ScanForFreeDevices* method you can scan for all free devices connected to the PC:

```
//Search for connected devices  
String[] devices = bodeAutomationInterface.ScanForFreeDevices();
```

Use the *ConnectWithSerialNumber* method to establish a connection to a Bode 100:

```
//Connect to the first found device  
BodeDevice device = bodeAutomationInterface.ConnectWithSerialNumber(devices[0]);
```

**Note:** Do not execute the above code directly in the GUI-Thread (this will cause a deadlock)!  
The connection can take up to 1.5 minutes when connecting the first time since the internal device calibration is executed during the very first connection.  
The following connections will only take roughly 1 second.

**Note:** You can also use Bode Analyzer Suite for the first connection. Then the internal device calibration will be done by the Bode Analyzer Suite. Future connects don't take so long.

### 3.2 Configure / Create a Measurement

```
//Create a S21 Measurement  
S21Measurement s21Measurement = device.Transmission.CreateS21Measurement();
```

```
//Configures a frequency sweep measurement  
s21Measurement.ConfigureSweep(1000, 10000000, 201, SweepMode.Linear);
```

```
//set source level  
s21Measurement.SetSourceLevel(0, LevelUnit.dBm);
```

**Note:** There are more possible measurements such as Impedance - One-Port etc.  
Furthermore there are additional settings such as attenuator configuration and termination.

### 3.3 Execute the Measurement

```
//Execute the measurement  
ExecutionState state = s21Measurement.ExecuteMeasurement();
```

**Note:** Execution returns the *ExecutionState* enum that can be “OK”, “Overload”, “Error”, “DeviceLost”, “Cancelled” or “CalibrationMandatory”.

### 3.4 Read the Results

```
//Read out the frequencies and measured magnitude values  
double[] frequencies = s21Measurement.Results.MeasurementFrequencies;  
double[] magnitudes = s21Measurement.Results.Magnitude(MagnitudeUnit.dB);
```

### 3.5 Disconnect from the Device

```
//Disconnect Bode 100 device to release it for future connection  
device.ShutDown();
```

## 4 Using Multiple Bode 100 Devices in Parallel

The methods shown in the previous section can also be used to control multiple Bode 100 devices from one computer. Controlling multiple devices from one computer adds additional complexity. The following chapter shall provide hints what to consider in such a case.

### 4.1 Hardware Considerations

Besides different software requirements, also new hardware requirements appear when using multiple Bode 100 devices in an automated environment.

#### 4.1.1 Measurement Setup

In many cases, paralleling is used to measure multiple DUTs that are manufactured on a batch. The batch can contain many DUTs that are located closely together and are therefore subject to EMI or electromagnetic coupling.

#### NOTICE

The measurement signal from one Bode 100 measurement setup can electromagnetically couple into a second Bode 100 measurement setup.

To avoid electromagnetic coupling from one measurement to another use shielded measurement connections and avoid creating geometrically big loop-antennas. The following picture shows a **problematic** example where the ground connection is routed around all devices under test:

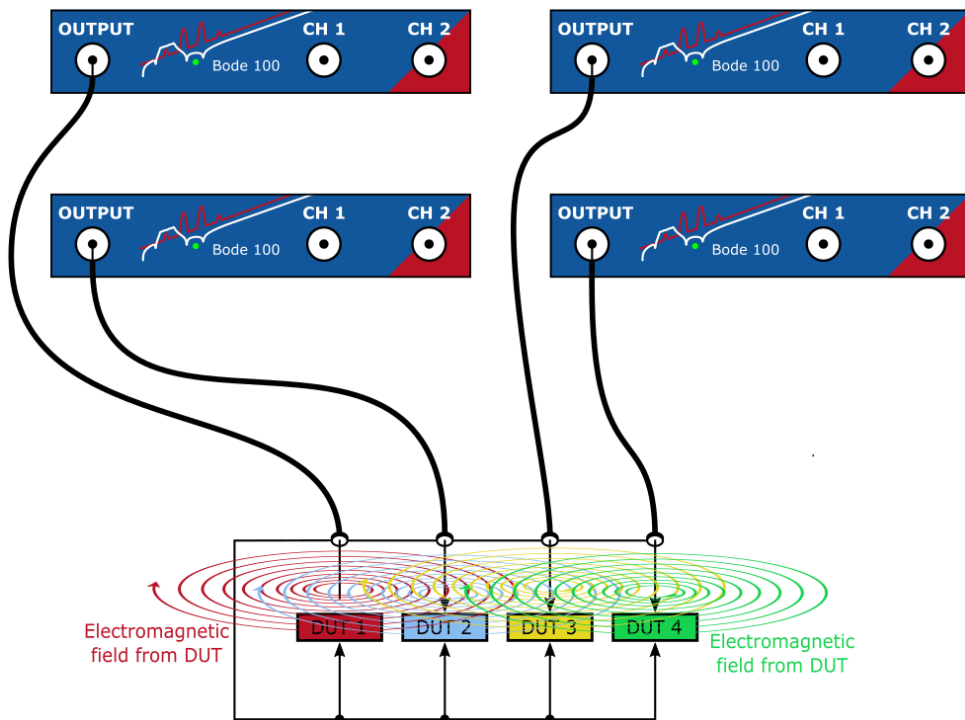


Figure 2: Problematic measurement setup, subject to electromagnetic coupling

It would be a lot better to keep the loops small and keep the return path of the signal close to the signal line, to reduce electromagnetic fields as shown in the following figure:

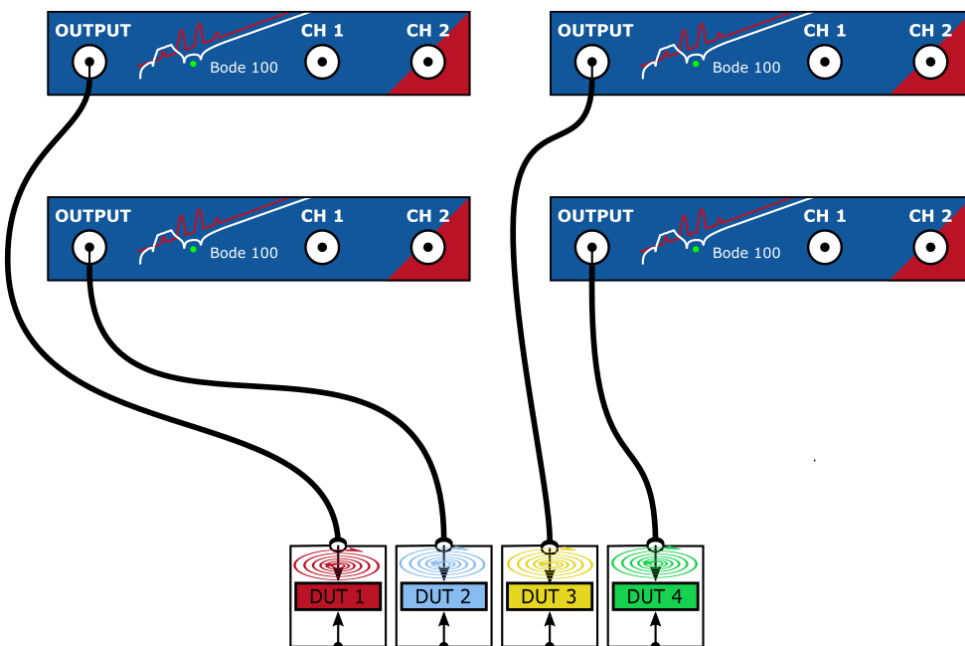


Figure 3: Better setup for reduced crosstalk. Loops are kept small.

In addition, or alternatively, you can add a time-delay to the different frequency sweeps to avoid measuring at the same frequency at the same time. The narrowband receivers of Bode 100 can easily reject disturbances at frequencies different to the measurement frequency.

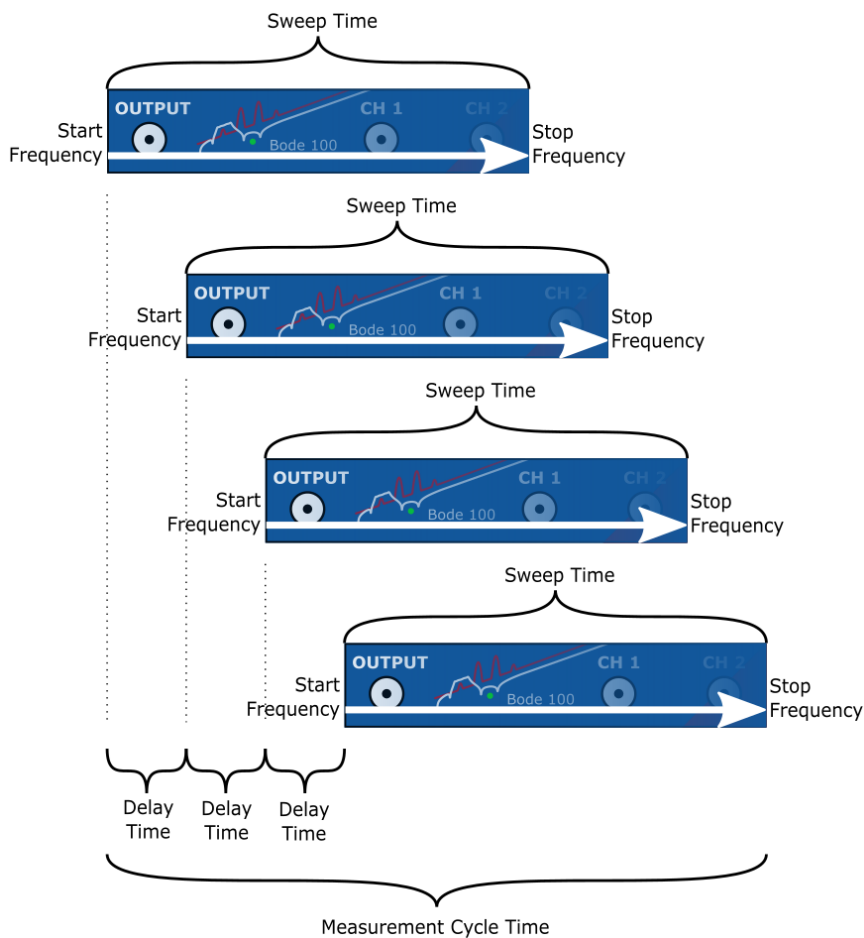


Figure 4: Avoid frequency-overlapping

This method will increase the total cycle time but provides excellent isolation between the different setups due to the high side-band rejection of Bode 100.

#### 4.1.2 USB Ports

It is of advantage to use a computer with one USB port per Bode 100 plus 2-3 ports for keyboard and mouse operation. If the computer does not provide enough ports, PCI express (PCIe) cards can be used to extend the number of ports on a PC. PCIe extension cards have the advantage that they appear as native USB root hubs on the system. This can overcome limitations on the system's root hub which can arise when using external USB hubs.

We have successfully tested the following PCIe card with 4 Bode 100 devices:

Sonnet Allegro 4-Port USB-A PCIe card (Fresco Logic FL1100EX Host Controller)

If you need to use an external USB hub, we recommend the following multi-TT device:

Elektron Overhub (7-Port USB 3.0 hub with multi transaction translator technology)



### 4.1.3 System Requirements

Since all the digital signal processing of Bode 100 is done on the computer, the processing power rises when more devices are connected to the computer. Furthermore, the receiver bandwidth setting strongly influences the processing power requirements:

Parameter	Effect
Number of devices	Higher number of devices → higher CPU requirements
Receiver bandwidth	Higher RBW value → higher CPU requirements
Number of points	Higher number of points → higher memory & CPU requirements

#### Selecting a Processor:

We recommend to select a processor that provides **one thread per Bode 100**. We have performed our tests using Intel Xeon processors that have 4 to 6 cores and 8 to 12 threads per processor.

The following tests have been performed at OMICRON Lab by doing a repeated frequency sweep from 30 kHz to 350 kHz (252 points). Note that **one process per Bode 100** device was used:

Processor	No. of Bode 100 Devices	Receiver Bandwidth	CPU Usage
Xeon <b>E3-1505M v6</b>	8	1 kHz ... 3 kHz	65 % ... 80 %
	8	100 Hz ... 300 Hz	25 % ... 50 %
	8	1 Hz ... 30 Hz	≤ 25 %
Xeon <b>E5-1650 v4</b>	10	1 kHz ... 3 kHz	40 % ... 56 %
	10	100 Hz ... 300 Hz	12 % ... 15 %
	10	1 Hz ... 30 Hz	≤ 12 %
Xeon <b>E5-1650 v4</b>	15	1 kHz ... 3 kHz	75 % ... 90 %
	15	100 Hz ... 300 Hz	20 % ... 25 %
	15	1 Hz ... 30 Hz	≤ 20 %

#### NOTICE

These are typical values! We recommend to keep enough processor margin!

Think about what your processor needs to do besides measuring with Bode 100. Do not plan to use more than ~ 50% of your processing power continuously.

Processor settings / turbo speed can influence these values.

### Selecting Memory:

For Microsoft Windows 10, 8 GB of memory are recommended. If you plan to measure several thousands of points in a sweep, please get in contact with us such that we can check your personal memory requirements.

## 4.2 Software Architecture

When using multiple devices on one computer, one can choose between two basic approaches:

- Single-process architecture
- Multi-process architecture

**Note:** When connecting to multiple devices you should always connect/initialize one device after another to avoid USB driver problems that can appear when connecting/opening multiple connections exactly at the same time.

### 4.2.1 Single-Process Architecture

The single-process architecture is easier to program and can be less stressful on the CPU or memory usage. But it has a higher potential for interference between the measurements, e.g. problems in one measurement could influence the execution performance of another measurement. Also, an unrecoverable error in one measurement would bring down all measurements including your software.

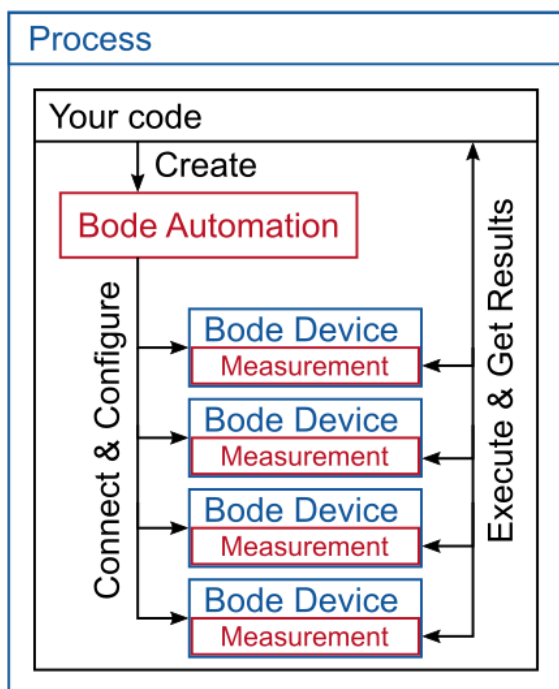


Figure 5: Single-process architecture

### 4.2.2 Multi-Process Architecture

A multi-process architecture offers advantages over the single process structure. One major advantage is the fact that every process that is responsible for a device can be monitored in software and restarted in case of an error. This can be used to ensure highest availability of your system. The disadvantage however, is a higher implementation effort and the need of inter-process-communication (IPC). Best practice would be to use one process per Bode 100 device as shown in the following figure.

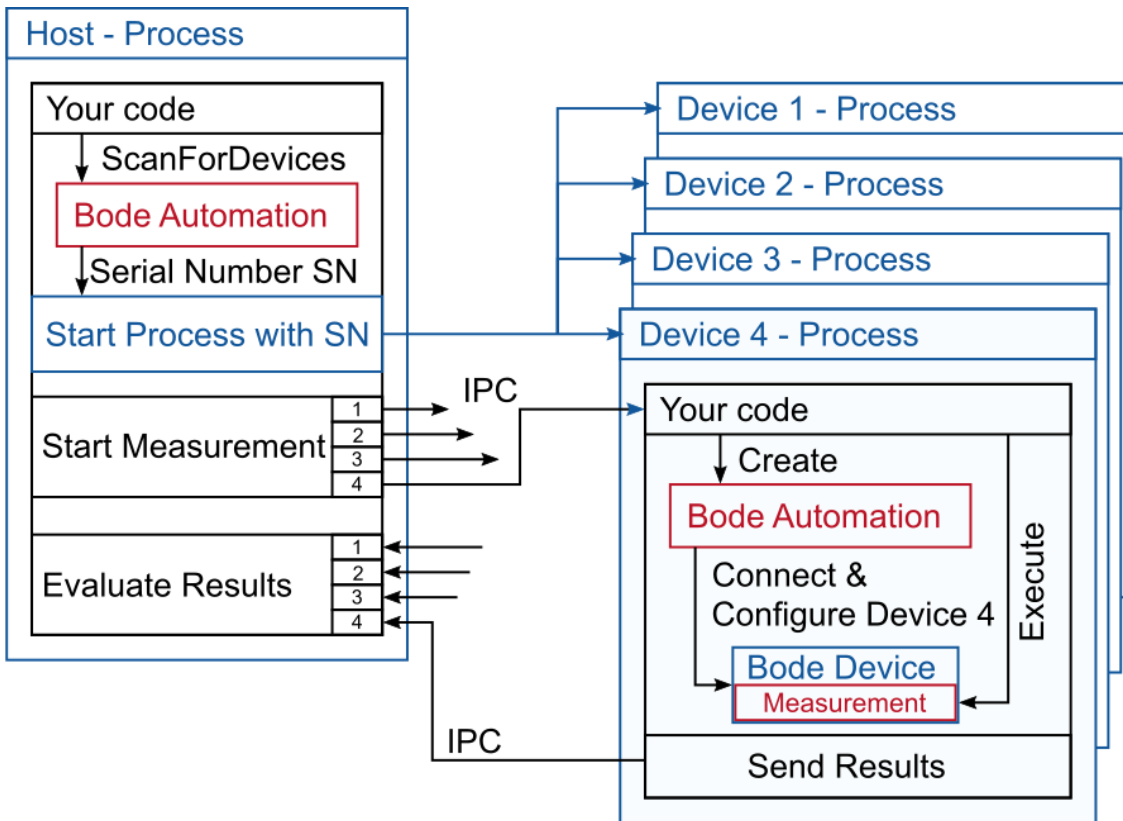


Figure 6: Multi-process architecture

**Note:** OMICRON Lab can provide a demo application on request. The demo is written in C#.NET and can showcase the basic principle of such an architecture.

## 5 Summary

Due to its multifunctional hardware design, the Bode 100 instrument fits perfectly to assure quality in production systems such as:

- LF and HF RFID Tag manufacturing
- Piezo element production
- Magnetics manufacturing
- Capacitor manufacturing

The wide frequency range of Bode 100 (1 Hz to 50 MHz) makes it easy to replace obsolete instruments such as the HP 4194A Impedance/Gain-Phase Analyzer or the E4991A in the lower frequency band. In addition, Bode 100 covers the full frequency range of the obsolete 4192A LF impedance analyzer.

Frequency sweep measurements can be performed quickly and the results can easily be evaluated directly at the automation computer. The USB interface offers a convenient way to connect the instrument to the automation computer.



**OMICRON Lab** is a division of OMICRON electronics GmbH, specialized in providing Smart Measurement Solutions to professionals such as scientists, engineers and teachers engaged in the field of electronics. It simplifies measurement tasks and provides its customers with more time to focus on their real business.

OMICRON Lab was established in 2006 and is meanwhile serving customers in more than 60 countries. Offices in America, Europe, East Asia and an international network of distributors enable a fast and extraordinary customer support.

OMICRON Lab products stand for high quality offered at the best price/value ratio on the market. The products' reliability and ease of use guarantee trouble-free operation. Close customer relationship and more than 30 years in-house experience enable the development of innovative products close to the field.

**Europe, Middle East, Africa**

OMICRON electronics GmbH

Phone: +43 59495

Fax: +43 59495 9999

**Asia Pacific**

OMICRON electronics Asia Limited

Phone: +852 3767 5500

**Americas**

OMICRON electronics Corp. USA

Phone: +1 713 830-4660