Mini-Circuits®

Test Solutions - Quick Setup Guide

This guide summarises the software resources, documentation and connection methods available for Mini-Circuits' rack-mount and modular test systems.

Contents

Contact

Please contact testsolutions@minicircuits.com if you have any questions or require further support.

Software Support / Download Website

Mini-Circuits' software and documentation resources are available for download from https://www.minicircuits.com/softwaredownload/software_download.html, select the model family from the list to access the relevant links. Since all Mini-Circuits' software controlled tests systems are based on the same interfaces, programming examples found here for other model families can usually also be applied to your product.

Supported Models

Product Family	Format	Supported Models
Modular Test Systems	Rack-Mount & Benchtop	ZTM, RCM
Programmable Attenuators (Single & 4-Channel)	Benchtop	RUDAT, RCDAT, RC4DAT
Programmable Attenuator Assemblies (Multi-Channel)	Rack-Mount	ZTDAT, ZTMN
Mechanical Switch Boxes	Benchtop	RC, USB
Solid State Switches	Benchtop	USB-1SP8T-63H, USB-4SP2T-63H, USB-2SP2T-DCH, U2C-1SP4T-63H, USB-SP4T-63
Solid-State Switch Assemblies	Rack-Mount	ZTS
2 x n Switch Matrices	Rack-Mount	ZTVX
Custom Switch Assemblies	Rack-Mount	ZT
Power Meters (Peak & Average)	Benchtop	PWR
Integrated Frequency Counter & Power Meters	Benchtop	FCPM-6000RC
Frequency Counters	Benchtop	UFC-6000
Signal Generators	Benchtop	SSG
USB to I/O Control Boxes	Benchtop	USB-I/O
RS232/USB to SPI Converters	Benchtop	RS232/USB-SPI, RS232/USB-SPI-N

USB Control

All products are configured using the USB HID (Human Interface Device) class so should automatically be recognized by the operating system. There is no need to install any additional drivers, simply plug in the USB connection and turn on the AC power supply. Control via USB is accomplished using the included GUI application software for Windows, the ActiveX / .NET DLL for programming in a Windows environment, or the USB interrupt API for programming in a Linux environment.

Ethernet Control

All products have DHCP enabled by default so a dynamic IP address should be automatically assigned by the network (where supported). The Ethernet configuration can be adjusted (for example to set a specific static IP address) using our GUI program, or programmatically using the API. In most cases a USB connection is required in order to change the Ethernet configuration.

Control via Ethernet is accomplished either using the included GUI (for Windows) or by sending simple text commands via HTTP or Telnet. These protocols are supported by the majority of programming environments and any web browser can be used as a simple console by entering the HTTP commands in the address bar. The example commands below can be sent programmatically or using a web browser to read the model name / serial number of a system:

`http://10.10.10.10/MN?`   // Read model name of unit with IP address 10.10.10.10
`http://10.10.10.10/SN?`   // Read serial number of unit with IP address 10.10.10.10

Software & Documentation Resources

The below software and documentation package is available for all models.

1. Graphical User Interface (GUI)
   - User friendly Windows application for manual control of the system via USB or Ethernet

2. Programming Manual
   - Full instructions for automating control through a variety of common programming environments:
     - USB control for Windows using the ActiveX / .NET DLLs
     - USB control for Linux systems using USB interrupt codes
     - Ethernet control using HTTP / Telnet (independent of operating system)

3. User Manual
   - Instructions on how to un-package, set up and get started with the system
   - Operating instructions for the Windows GUI application software

4. API DLL Files
   - ActiveX and .NET DLL files for user automation via USB (the .NET files have "64" at the end of the filename)
   - Supported by a wide variety of programming environments on Windows operating systems
   - ActiveX DLL: Unzip the file and refer to the ActiveX DLL File Registration (USB) section below
   - .NET DLL: Unzip the file and copy into your project folder (**no registration required**)

5. Programming Examples & Troubleshooting Guide
   - Worked examples for a number of common programming environments, including LabVIEW, Python, MatLab, C#, VB and others
   - Resolutions for common programming errors and DLL issues

6. FAQ (Frequently Asked Question) Application Notes
   - Application notes detailing common user queries and the responses
   - Guides are published on the website for:
     - Software / programming questions
     - Hardware / operation questions
     - LabVIEW specific questions

ActiveX DLL File Registration (USB)

This applies to the ActiveX DLL file only, .NET files do not need to be (and cannot be) registered.

1. Copy the ActiveX DLL file to the correct system directory listed below:
   - For 32-bit Windows operating systems this is `C:\WINDOWS\System32`
   - For 64-bit Windows operating systems this is `C:\WINDOWS\SysWOW64`

2. Open the Command Prompt:
   a) For Windows XP® (see Fig 1):
      i. Select "All Programs", "Accessories", "Command Prompt" from the Start Menu
   b) For later versions of Windows the Command Prompt must be opened in "Elevated" mode:
      i. Open the Start Menu/Start Screen and type "Command Prompt"
      ii. Right-click on the shortcut for the Command Prompt and select "Run as Administrator"

3. Use regsvr32 to register the DLL:
   - For 32-bit Windows operating systems type (see Fig 2):
      o `\WINDOWS\System32\Regsvr32 \WINDOWS\System32\[filename].dll`
   - For 64-bit Windows operating systems type (see Fig 2):
      o `\WINDOWS\SysWOW64\Regsvr32 \WINDOWS\SysWOW64\[filename].dll`

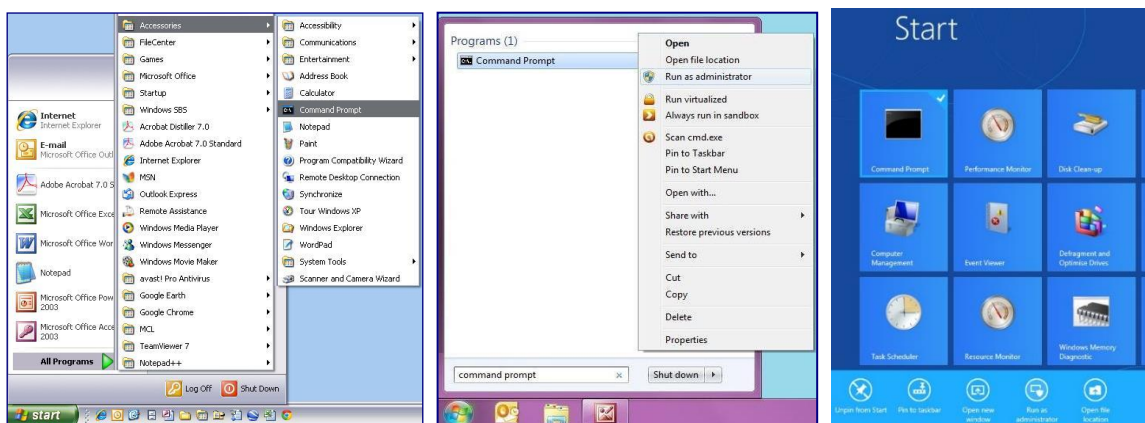4. Hit enter and a message box will appear to inform that registration was successful.

Fig 1 - Opening the Command Prompt in Windows XP (left), Windows 7 (middle) and Windows 8 (right)
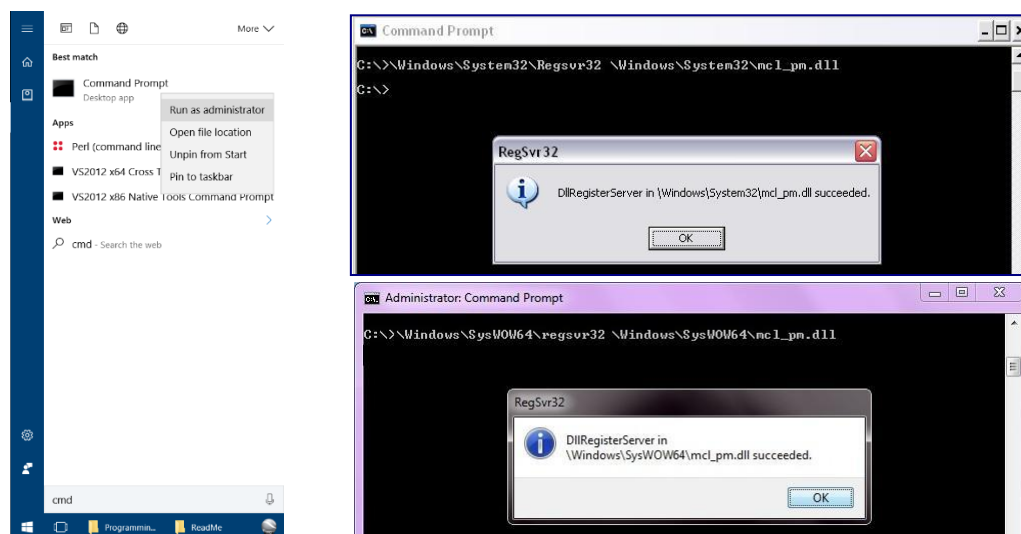

Fig 2 - Opening the Command Prompt in Windows 10 (left); registering the DLL (32-bit top left, 64-bit bottom left)