

Controlling LXI Instrumentation with Smart Devices

By [Neil Forcier](#)
[Agilent Technologies](#)

You are on your way to catch a flight to India where you will help an important customer integrate your company's solar-powered base station into a rural area near Bangalore. As you board the plane, you want to check the status of ongoing environmental tests being performed in your lab on an updated base station especially designed for hot and humid rural climates. You whip out your iPhone and connect to your Agilent 34972A LXI data acquisition unit and read the results from the most recent temperature scans. Everything checks out, so you raise the temperature of the chamber via the digital control module on the DAQ unit. You power down your smart device for take-off and settle in for a long flight.

Is it a reality or a pipedream?

Introduction

The LXI (LAN eXtensions for Instrumentation) instrument I/O standard was built upon the ubiquitous Ethernet or LAN network standard, the same standard that makes the Internet possible. Because of its Ethernet foundation, one big advantage LXI offers is the flexibility engendered by distance between a computer or controller and the LXI instrument it is connected to.

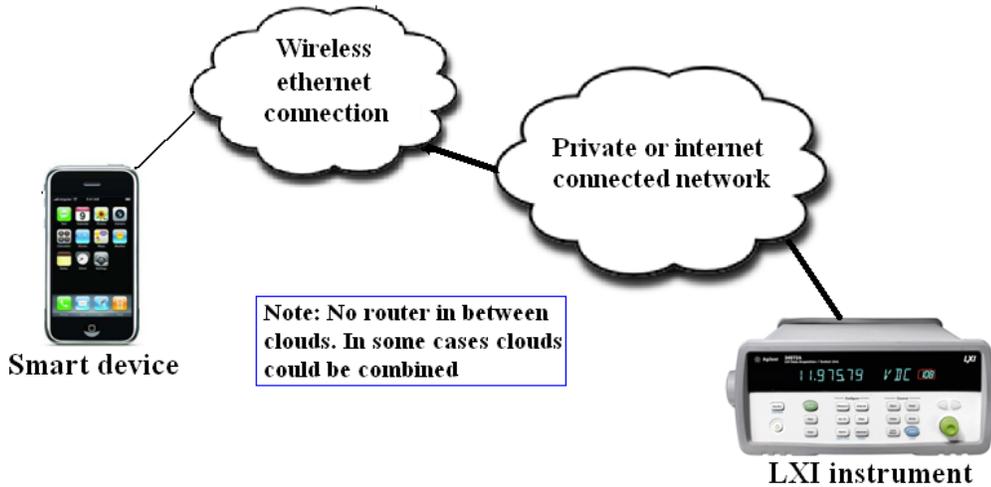
Smart devices, including smart phones and tablet PCs, are rapidly becoming pervasive. At this point, no one can say with certainty what impact these devices will have on the test and measurement industry. These smart devices are a natural fit with the LXI standard since they can communicate with an Ethernet network, either using Wi-Fi or through a cellular provider's network via the Internet. The wireless freedom of smart devices further increases the remote flexibility of LXI instrumentation.

In this article we will explore how you can use smart devices to communicate with LXI instruments from the network level down to the code level. We will offer insights into the extra time required and the complexities you will face, at this point in smart device evolution, to access the advantages they provide for controlling LXI instrumentation remotely. The extra time and added complexities arise from network security obstacles that you will encounter when using your smart device remotely. Also, smart devices are a new remote instrument control platform, so there is not much software and not many programming tools available to assist you. Finally, we will provide you with access to free code to get you started controlling instruments remotely on the iOS platform (iPhone, iPod, iPad).

Communicating at the Network Level

There are three different network hierarchies for using a smart device to connect to an LXI instrument that I will call Wi-Fi local network, Wi-Fi non-local network, and wireless cellular network. We will start with the easiest but least flexible hierarchy, the local network. In this configuration, the smart device is connected via Wi-Fi to the same local network that the LXI instrument is connected to, which means there are no firewalls or network security barriers between them. This type of connection is shown in Figure 1.

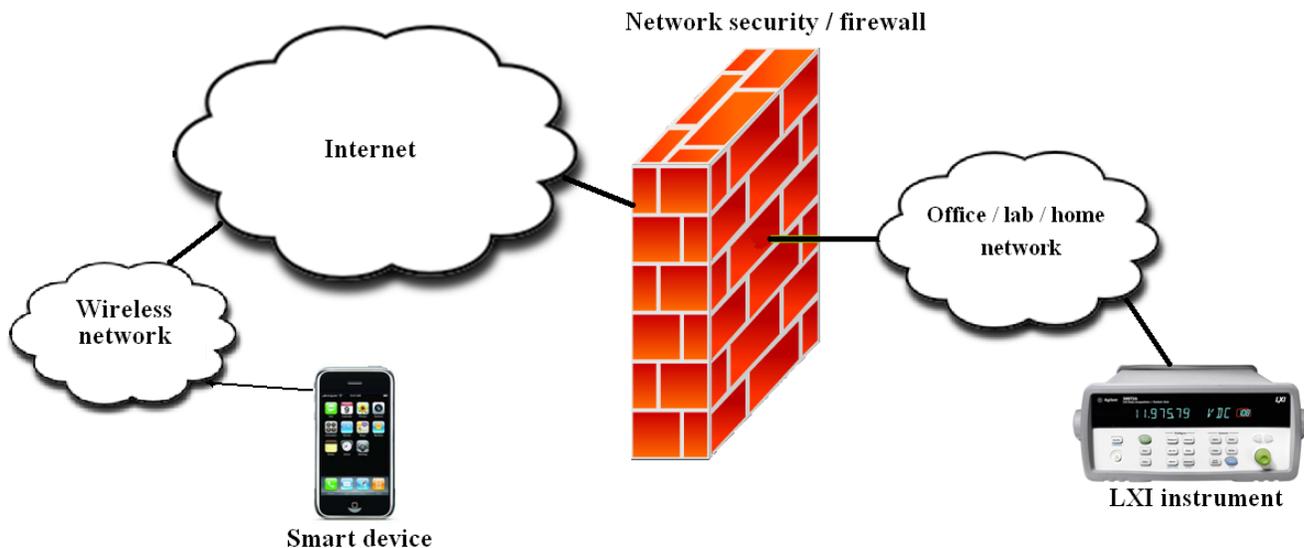
Figure 1. Connecting via local network



There are no network security blocks in the way, so it is easy to make a connection to the instrument. You can test this method easily using the LXI instrument's built-in Web server. Simply connect to the local Wi-Fi network with your smart device, obtain the IP address of an LXI instrument connected to the same local network, open the Web browser on your smart device, enter the IP address of the instrument and press "go." You should see the instrument's introductory Web page pop-up. Of course, connecting to the instrument via a local Wi-Fi network does not provide much distance flexibility since Wi-Fi network range is typically limited to 32 meters indoors and 95 meters outdoors.

The second network hierarchy, Wi-Fi non-local network, involves connecting to a Wi-Fi network with your smart device and reaching across the cloud to the local network where your LXI instrument is connected. The challenge this hierarchy presents is getting through network security measures such as firewalls. The Wi-Fi non-local network hierarchy is shown in Figure 2.

Figure 2. Wi-Fi non-local network



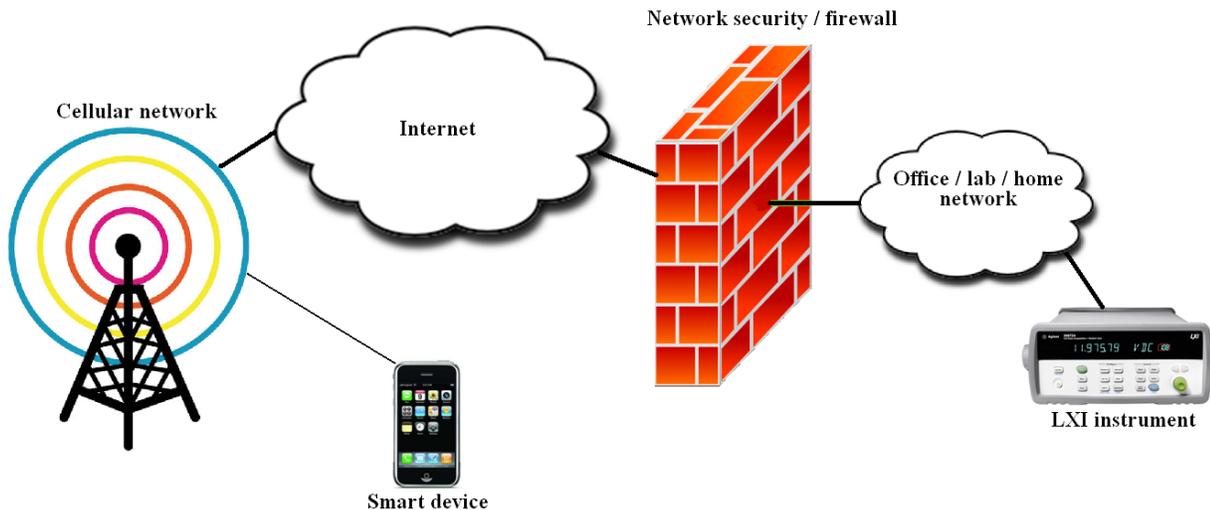
You encounter this type of hierarchy when you try to connect to an instrument in the office or lab from your home wireless network.

Let's take the simplest case first: Suppose you are fortunate enough to work from home and you have your LXI instrument connected to your home router, which is connected to the Internet. By default, your router is configured to let connection requests leave your local network but not come in. Network software assumes everything on your local network is a client, not a server, and therefore it blocks incoming connection requests to protect the machines on your local network. Your LXI instrument is a server, so when your smart device tries to connect to it from across the cloud, your router will block it. To overcome this problem, use the port forwarding or DMZ (demilitarized zone) capability on your router. Port forwarding tells your router to forward all network traffic on a particular port or ports that you define to a local IP address on your network (in this case your LXI instrument). With DMZ capability, your router makes a specified machine on your local network appear as if it is directly connected to the Internet. I do not recommend the DMZ method for any instruments running Windows® operating system, because it leaves them wide open to cyber attacks.

For most of us, our LXI test and measurement instruments are on a company or organization network that is heavily fortified from the Internet. In this case, you will need company- or organization-approved virtual private network (VPN) software on your smart device to gain access to your office network. Also, you would want to consult with your IT department in case your network is configured to restrict certain types of incoming network traffic that encompasses your smart device.

Finally, there is the third hierarchy using your smart device's wireless provider's network to connect across the cloud to your instrument, as shown in Figure 3.

Figure 3. Connecting across wireless provider's network



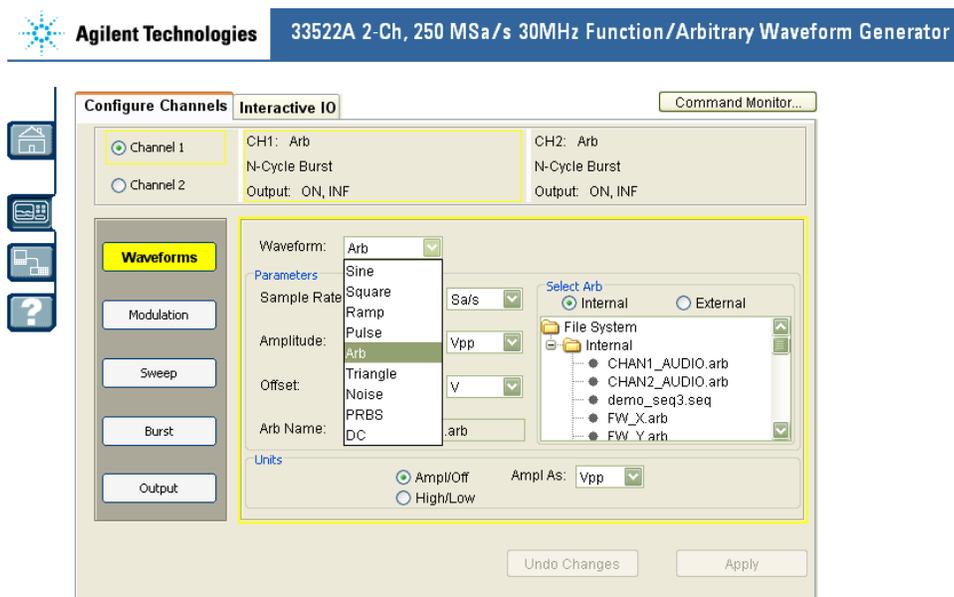
In this scenario, we face the same network security challenges we encountered in the previous hierarchy. The only difference between the two is using the cellular provider's network allows greater remote flexibility because you no longer are restricted by the range of typical Wi-Fi networks, but your data bandwidth is limited compared to a Wi-Fi connection. Large bandwidth is not really needed in this remote instrument use model. Since smart devices are used on the go, you cannot guarantee a constant connection between you and your instrumentation, so you can only use it as a measurement monitoring tool and to make simple non-critical control decisions. Data traffic that consists of only measurement readings and simple control SCPI commands does not require much bandwidth.

Communicating at the Code Level

So far, we have covered the various ways you can connect a smart device and communicate with an LXI instrument from a network standpoint. One important piece of the puzzle is the software that tells the hardware and network what to do to facilitate the communication.

Unfortunately there currently are no apps for connecting and communicating with LXI instrumentation with smart devices. The LXI standard requires each LXI instrument to have a built-in server that can host a Web interface that can be accessed via a Web browser. In its most basic form, the Web interface must provide users access to the LAN settings of the instruments. Most test and measurement vendors have gone the extra step and added instrument control capability to the Web page. You can see an example of an LXI Web control page from Agilent's 33522A function/arbitrary waveform generator in Figure 4.

Figure 4. 33522A Web graphical user interface



Smart devices have Web browsers so you can use them to connect and control LXI instruments that provide control capability, right? The connection part works fine when you use a smart device's Web browser. You enter the IP address into the browser, hit go, and you see the instruments introduction page show up in the browser. The problem is that the majority of instrument control Web interfaces require some type of plug-in, typically Java, to access the control user interface. These plug-ins are currently not available on smart device operating systems, which means using the instrument's LXI Web pages to control the instrument won't work with a smart device in most cases.

That leaves us only one choice if we want to use our smart device to control our LXI test and measurement instrument: create the software ourselves. When you create software on a Windows-based PC, many tools are available, such as VISA and IVI-COM drivers, that make instrument programming easier by doing all the low-level connection management and data handling for you.

As of the writing of this article, the only way to connect and communicate with an LXI instrument using a smart device is to use network sockets. If you are unfamiliar with network sockets, check out Wikipedia's internet socket page at: http://en.wikipedia.org/wiki/Internet_socket.

Anyone who has worked with sockets knows network programming is not easy. When you use drivers,

you don't have to worry about some of the programming challenges that low-level sockets normally present. For example, you do not have to worry about managing connections and formatting incoming and outgoing data when you use drivers.

There are many aspects to managing a connection to an instrument, but one of the trickiest is dealing with methods that block, or stop execution, until the method completes. I will use the most well-known socket API, the C BSD socket API, as an example. In BSD the connect() and recv() methods in their default state will block until they get a reply. If the device your program is communicating with loses its network connection and you call the connect() or recv() function, your program will stop execution with no way to recover other than restarting it. Obviously, this is undesirable. There are multiple ways to prevent these functions from blocking, such as setting up a timeout or executing them on a separate thread. Of course all this adds complexity to your software development.

Data handling is another aspect of using low-level sockets that adds time and complexity to programming. For example, you need to be able to handle data streams of various lengths from the instrument. The data comes in strings, so you'll need to convert to ints and floats at your desired precision level. Some of the smart device development environments provide tools that add levels of abstraction to low-level sockets. These tools can make managing a connection or handling data easier. But keep mind when using these tools that they are mainly created to connect and interact with internet Web sites, not test and measurement instrumentation.

If you go to my GPETE blog (<http://gpete-neil.blogspot.com/>), you will find an iOS program file entitled "starIDN." The program is made for the iPhone. StarIDN allows you to connect to an instrument using an IP address. Send a "*IDN" SCPI command, and it reads and displays the result. The code uses C BSD sockets to communicate with the instrument. Feel free to use the socket class as a starting point for your own smart device test and measurement app.

Conclusion

Smart devices offer you a portable and interactive way to access information and data wirelessly from almost anywhere. The main way smart devices access information and data is through Ethernet using a provider's cellular network or a Wi-Fi connection. Since the LXI connection standard is Ethernet based, LXI is a natural fit for monitoring and controlling test and measurement instrumentation with smart devices. Using smart devices to remotely monitor and control test and measurement instrumentation provides unmatched location flexibility. It allows you to access instrumentation from almost anywhere so you are no longer tied to a desk or lab during long tests.

At the time this article was written, there was no software or any drivers for using smart devices to control instrumentation. This means the only way to use a smart device to monitor and control test and measurement instruments is to get down and dirty with the code yourself. One thing you can count on: As smart devices continue to become a bigger part of our digital and wireless lives, they will find their way into the test and measurement industry.

Windows is a U.S. registered trademark of Microsoft Corporation.