MONITORING AND CONTROLLING AUTOMATION SYSTEMS USING SMARTPHONES / PDA

Daniel SORA

Regional Department of Defense Resources Management Studies

***Abstract**: Many people will be able to leave their laptop at the office and handle essentially all of their mobile computing and communications tasks with a pocket-sized device. Now, the smartphone might not be one's first choice for spreadsheets and documents, but let's face it, everybody has one (in business, anyway, all over the world), and those that have one will buy a new one sometime in the next two years as cellular contracts expire and products and wireless technologies continue their rapid evolution. Smartphones have been as powerful as PCs for just a few years, with significantly better software, user interfaces, and flexibility. The level of capability in contemporary smartphones is remarkable and continues to grow.*

***Keywords**: Smartphone, PDA, automation, monitoring, wireless, LabView, security*

1. OVERVIEW

The growth of wireless communication in the past few years means you can stay connected to a network regardless of whether you are in your neighborhood coffee shop or across the country. No longer bound by the harnesses of wired networks, you can access and share information on a global scale. PDA devices and other mobile handheld devices make it easier than ever to develop remote applications that transmit and receive information from a remote site back to a host computer [2].

LabVIEW (short for Laboratory Virtual Instrumentation Engineering Workbench) is a platform and development environment for a visual programming language from National Instruments. Originally released for the Apple Macintosh in 1986, LabVIEW is commonly used for data acquisition, instrument control, and industrial automation on a variety of platforms including Microsoft Windows, various flavors of UNIX, Linux, and Mac OS X. The latest version of LabVIEW is version LabVIEW 2009, released in August 2009 [3].

LabVIEW supports the Open System Interconnection (OSI) Model, so implementing wireless network communication in LabVIEW is very similar to implementing wired network communication.

LabVIEW programs – VIs (Virtual Instruments) can communicate, or network, with other processes, including those that run on other applications or on remote computers to perform the following tasks:

- Share live data with other VIs running on a network using shared variables.
- Publish front panel images and VI documentation on the Web.
- Email data from VIs.
- Build VIs that communicate with other applications and VIs through low-level protocols, such as TCP, UDP, Apple events, and PPC Toolbox.

2. LabVIEW AS A NETWORK CLIENT AND SERVER

You can use LabVIEW as a client to subscribe to data and use features in other applications or as a server to make LabVIEW features available to other applications.

Before you can access the properties and invoke methods of another application, you must establish the network protocol you use to access the properties and methods. Protocols you can use include HTTP and TCP/IP. The protocol you select depends on the application. For example, the HTTP protocol is ideal for publishing on the Web, but you cannot use the HTTP protocol to build a VI that listens for data that another VI creates. To do that, use the TCP protocol. LabVIEW supports several low-level protocols you can use to communicate between computers [4].

You can use ActiveX technology with LabVIEW as an ActiveX server or client.

Shared variables are configured software items that can send data between VIs. Use shared variables to share data among VIs or between locations in an application that cannot be connected with wires. A shared variable can represent a value or an I/O point. You can change the properties of a shared variable without having to edit the block diagram of the VIs that use the shared variable.

Network-published shared variables communicate between VIs, remote computers, and hardware through the Shared Variable Engine. The Shared Variable Engine uses the NI Publish-Subscribe Protocol (NI-PSP) data transfer protocol to write and allow users to read live data. NI-PSP is a proprietary technology that provides fast and reliable data transmission for large and small applications and is installed as a service on the computer when you install LabVIEW.

The NI-PSP networking protocol uses psp URLs to transmit data across the network. You can browse to any NI-PSP data item on the network to seamlessly bind shared variables to other shared variables or to server and device data items.

An NI-PSP data item can be a shared variable in a LabVIEW project other than an active project or a data item on a connected server or device, such as an OPC server or FieldPoint module.

3. CREATING SHARED VARIABLES USING LabVIEW PDA MODULE

Use shared variables to share data among VIs in the same PDA or Touch Panel application or to read data from and write data to other network-published shared variables. The shared variables provide a memory space that can be used to send and receive data between different targets in the project. One target in the project must host the shared variables. All other targets can connect to that host as clients to publish or subscribe to the data stored in the shared variable memory space.

The ***Project Explorer*** window provides the framework for organizing and interacting with various distributed targets from a single location within the LabVIEW programming environment. Shared variables provide an easy method for sharing data between various targets.

A shared variable is accessible through a network, but hosted on a single machine. When developing your VI in LabVIEW, the shared variable library is automatically deployed to the target it is listed under in the project explorer. When the library is deployed to this target, that target is now "hosting" the shared variables contained in the library. Other computers will have to connect to this target to read or write the shared variable library.

The PDA Module do not support the DataSocket Transport Protocol (DSTP) or hosting of network-published shared variables, which means you only can read and write to targets other than PDA targets.

To create a shared variable that runs on a PDA or Touch Panel target, right-click the target in the Project Explorer window and select ***New»Variable*** from the shortcut menu to open the Shared Variable Properties dialog box. After you configure the shared variable and click the **OK** button, LabVIEW creates a project library that contains the shared variable under the PDA or Touch Panel target.

After you create the shared variable, you can right-click the variable and select ***Properties*** from the shortcut menu to display the ***Shared Variable Properties*** dialog box. Use the ***Shared Variable Properties*** dialog box to configure the shared variable. The shared variable can be used to read and write shared variable values on the block diagram.

With the network-published shared variables hosted on the development PC in the project, you can use these shared variables to send and receive data between all of the targets in the project. The next step is to create applications that run on the host PC and the PDA target that publishes and subscribes to these shared variables.

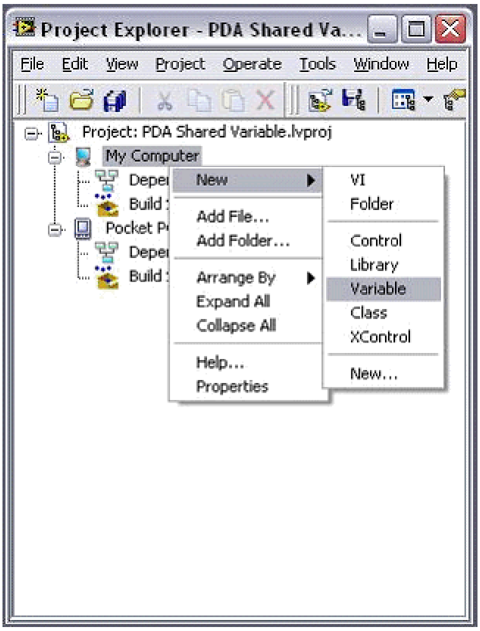

Fig. 1. Creating a new shared variable in the Project Explorer window

Wireless PDA applications consist of a client and a service. The client is the PDA application that monitors or controls remote services over the network and communicates directly with a service on a server or another device. Services are VIs or other applications that perform tasks that the client accesses. For example, a service might be a VI that monitors temperature or tank level. The client might be an application on a PDA target that uses the TCP Open Connection function to connect to a remote service on the network and then uses the TCP Read function to read the temperature or tank level data that the remote service broadcasts over TCP [5].

Note that the PDA device must have an active connection to the Ethernet or wireless network to subscribe to the network-published shared variables hosted on the host PC.

4. WIRELESS SECURITY IN PDA MONITORING AND CONTROLLING APPLICATIONS

NI Wi-Fi data acquisition (DAQ) devices use IEEE 802.11 to stream continuous waveform data over a wireless network. Because IEEE 802.11 uses over-the-air RF signals as its physical transmission medium, it offers unique security challenges beyond those of a wired system [6].

NI Wi-Fi DAQ supports the highest commercially available security, IEEE 802.11i (commonly known as WPA2 Enterprise).

For effective protection of wireless data transmissions, a Wi-Fi network must have a strong encryption algorithm (cipher) and some form of key management. Two encryption standards are widely used today with Wi-Fi networks: TKIP and AES.

The IEEE 802.11i task group introduced the Temporal Key Integrity Protocol (TKIP) with WPA as a stop gap for existing WEP networks. One advantage of TKIP over WEP is that it uses a 128-bit key versus a 40-bit key, though the encryption algorithm (RC4) is still the same. The more significant difference is that TKIP uses a different key for every message packet, hence the name “temporal.” This key is created dynamically by mixing a known pairwise transient key (PTK) with the MAC address of the client and a serial number for each packet. The PTK is created when a client connects to an access point using a preshared key (a passphrase that is known to all network members) and a random number generator. The serial number is incremented each time a new packet is sent. This means that replay attacks are impossible because the same key is never used from one packet to the next. An access point can detect when an attacker attempts to replay old packets.

As a final security solution, the IEEE 802.11i task group chose the Advanced Encryption Standard (AES) as the preferred encryption algorithm for Wi-Fi networks. AES uses a 128-bit cipher that is significantly more difficult to crack than the RC4 algorithm used by TKIP and WEP. In fact, the National Institute of Standards and Technology (NIST) chose AES as the encryption standard recommended for all U.S. government agencies. Any wireless data acquisition application for the government or military likely has to use AES to transmit data.

Authentication is the second key component of wireless security. Network authentication is essentially client access control. Before a client can communicate with a wireless access point, it must authenticate with the network. There are two basic forms of authentication: server-based and preshared key (PSK)-based.

A successful authentication process results in a pairwise master key (PMK) used to encrypt wireless traffic. The details of this exchange depend on which Extensible Authentication Protocol (EAP) method the network supports.

NI Wi-Fi DAQ devices support the full IEEE 802.11i security standard, including AES encryption and IEEE 802.1X authentication. This is the highest commercially available wireless network security, meaning your sensitive data is protected from unwanted access.

Fig. 2. NI Wi-Fi DAQ streams continuous waveform data over a secure IEEE 802.11 network.

Security settings for NI Wi-Fi DAQ devices are easy to use. In Measurement & Automation Explorer (MAX), select your NI Wi-Fi DAQ device under “NI-DAQmx Devices” and click on the “Network” tab at the bottom of the screen. Select the “Wireless” tab to configure your network security options with a series of drop-down menus.

If your EAP method requires a client-side certificate, be sure to obtain it before attempting to set up your data acquisition device. And if you are setting up your own network without an authentication server, be sure to use a strong PSK passphrase (with both WPA and WPA2 networks).

5. CONCLUSION

The LabVIEW PDA Module provides a set of tools to facilitate the implementation of security for local or network resources such as LabVIEW project libraries, shared variables, and front panel objects. Project libraries can be used as containers to assign permissions to multiple shared variables or other project libraries contained in that project library. Although shared variables inherit permissions from the project library where they reside, shared variables can assign individual permissions as well.

REFERENCES

[1] Craig Mathias, *"Can the smartphone replace the laptop?"* - MOBILE TECHNOLOGIES AND TRENDS - 08.10.2009

[2] http://zone.ni.com/devzone/cda/tut/p/id/3280 *"Developing Wireless PDA Applications in LabVIEW"*

[3] http://en.wikipedia.org/wiki/LabVIEW

[4] http://zone.ni.com/reference/en-XX/help/371361B-01/TOC27.htm "*Networking in LabVIEW*"

[5] http://zone.ni.com/devzone/cda/tut/p/id/4375 "*Creating and Using Shared Variables in the LabVIEW 8.20 PDA Module*"

[6] http://zone.ni.com/devzone/cda/tut/p/id/7376 "*Wireless Security Primer for Data Acquisition*"