Motivation

As wireless technology improves, devices with wireless capability have become a regular part of our lives. After iPhone, BlackBerry, and Android phones became available on the market, and more and more applications have been developed. However, most of the phone applications developed so far are used primarily for entertainment purposes. There are currently no phone applications developed for interfacing industrial control systems although industrial control systems that can take advantage of wireless technology exist. For example, in a typical industrial control system, the plant operator, engineer or manager must use the control system consoles to monitor real-time data of the field devices (like motor, pump, and valve). If the plant personnel are not at a control console then they cannot review the system information. This is very inconvenient for all of the employees. If they had a “mobile console” implemented in their cell phones then they could check the control system status at anywhere and anytime.

In this project we developed an Android phone application (using Android SDK) which can directly communicate with field device controllers using an industrial control protocol. The application has the features of retrieving real-time device data, receiving critical alarms from field devices, and sending commands to acknowledge alarms or control field devices. The cell phone connects to the control system TCP/IP network via a Wi-Fi router (a Wireless Access Point).

The cell phone application implements a popular control system protocol called Modbus to send commands to and receive data from the field device controller using TCP sockets. The field device controller used in this project is a Programmable Logic Controller (PLC). PLC is the main workhorse in the industrial automation control systems and is a microprocessor based control device which can be programmed to monitor and control industrial devices. PLC has I/O modules which are used to collect field device signals and actuate the devices. Most the PLCs in the control systems support Modbus protocol.

Goals

The goals of this project were:

1. To read real-time data (several analog inputs and several discrete inputs) from an Allen-Bradley compactLogix PLC based industrial control system and display the real-time values on an Android cell phone.
2. Send commands from the cell phone application to the PLC’s relay output to actuate a simulated field device (For example, send a command from the cell phone to start/stop a pump or open/close a valve.)
3. Develop an intuitive graphic user interface that will allow the operator to easily utilize all of the functions available in the program.
4. Develop a graphical status monitor, to include animation, in order to display the data in the most visual manner possible.

**Project Diagram**

The following graph demonstrates the network diagram of this project. It shows the hardware required in this project and the dataflow in the communication.

![Project Diagram Image]

**Hardware**
We use an Allen Bradley CompactLogix PLC as the industrial controller. We use a Linksys WRT54G wireless router to simulate the industrial plant Wi-Fi connection. We use a HTC Dream phone (Android Development Phone) as the mobile device to interface with the PLC.

![Hardware Image]

**Software**
We have developed a PLC ladder logic program (using RSLogix 5000 software) for communicating with an Android phone via Modbus protocol. The PLC program collects field sensor data and feeds the data to the Android phone. We have implemented some basic Modbus commands in Android such as read coils (Function code 1), read input registers (Function code 4), and write coils (Function code 15). During our development and testing, we used Wireshark (a network packet analyzer) to display the Modbus TCP packets between the Android phone and the PLC. With TCP packets intercepted by Wireshark we were able to verify the correctness of the packets we sent out and the packets we received.

We have developed the cyclic redundancy check algorithm (CRC16) in Java. In Modbus communication, the sender calculates a 16-bit value (checksum) based on every byte in the packet using the CRC16 algorithm. After the receiver receives the packet it also calculates the checksum using the same CRC16 algorithm. If the checksums match then the receiver will confirm the received packet is good otherwise the receiver will discard the packet because transmission error has been detected.
We designed the following screens as our GUI interface.
**Functionality**

Our program allows for two-way communication between the devices in a way that is time sensitive and continuous.

The first screen demonstrates communication from the PLC to the phone. In this case, on/off status and alarms of several devices are displayed. Should a status change occur in a field device, an alarm consisting of sound and vibration will alert the operator.

The second screen demonstrates communication from the PLC in the form of analog values. The devices tied to it are a water level gauge (labeled as 30001) and a digital thermometer (labeled as 30002). The final value (labeled as 30003) displayed is the “heartbeat” monitor of the wireless communication link. This is a protection against communications failure. The PLC will update its “heartbeat” counter once a second. When the phone polls the PLC, it will check the counter to ensure that it is higher than the previous one received. Should the value be the same, this is a sign that the data it is displaying is old data (not real-time data). In this way, we ensure that the information displayed on the phone is the most current data (real-time data) and the operator does not send any command to the field device accidently due to the obsolete data displayed on the phone.

The third screen is interactive in that it allows the user to control the actuators by sending commands to the PLC. It also displays the status of the devices in the same way that was demonstrated in screen 1.

The fourth screen uses an animated graphic to demonstrate a typical closed-loop control scenario (Pump-Tank control). When the water level in the tank is below the low set point the pump will run to fill the tank. When the water level in the tank is above the high set point the pump will stop filling the tank. This screen is also used for a small simulation program that we built in order to test autonomous operation. The water level will continuously rise and fall, triggering the pump to start or stop accordingly. This animated simulation demonstrates how the program can react to the field device based on its status change.

**Steps to development**

While we accomplished what we set out to do, there are several features that should be included before such a system would be ready for deployment. The first of these is scalability. Currently, only one phone can talk to one PLC at a time. It would be appropriate for the program to allow for several phones to talk to one PLC and for one phone to talk to several PLCs. The second major issue is with security. Currently we employ no authentication, permissions or encryptions. It is relatively easy to see how such a system could be compromised without these three things, particularly in a critical industrial application.

**Issues with wireless technology**

We chose the 802.11b wireless protocol for this system due to its ease of use. However, it could easily be tied into another protocol such as cellular if need be. There are strengths and weaknesses to all wireless
technologies but some issues remain constant throughout. All wireless frequencies are generally less reliable than wired. This is primarily due to the weakening of the signal over distances and the increase in the number of factors that can cause interference. While wired technologies are relatively isolated, wireless is affected by all forms of matter, other electronic signals being broadcasted, and even the atmosphere that it must travel through. There is also a much greater security risk due to the fact that the signal can be more easily intercepted by anyone in the area. Even worse, a signal can be sent from an outside source in order to influence the receiving device. These are all serious concerns that must be addressed through increased and multi-layered security practices and technologies before wireless can be used on critical systems, particularly those that could influence a human life. So in the wireless application design we must consider such impacts. We need to come up with a strategy (algorithm) to handle the worst case scenario (in both value domain and time domain) and therefore the application will continue to provide the functions that it’s designed for (maybe in a downgrade level).

Conclusion

We (probably the first who) successfully designed and implemented an Android phone application (we call it gRemote) which can interface with an industrial control system for both monitor and control. The application can read real-time sensor data and receive alarms from the field devices and can send commands to acknowledge alarms or control the devices. This application will significantly change the way that the industrial operators monitor and control their systems. Although efforts have to be made to address the issues of security and determinism in such applications, we estimate that the wireless interface for the industrial control systems will become a standard feature in the next few years.