Remote Wireless Control of a Bottling Process

DAVID HERGERT, Ph.D.
Professor, Engineering Technology
Miami University-Hamilton
1601 University Blvd.
Hamilton Ohio 45011
hergerd@muohio.edu

Remote Wireless Control of a Bottling Process

Abstract:
Over the last ten years, remote wireless monitoring and control has become an integral part of industrial automation systems. Remote monitoring is used in such diverse areas as automobile assembly, oil and process control, analyzing temperature in heat exchangers, deployment of resources on a smart grid, and environmental measurements.
This paper describes a remote wireless monitoring and control system used in a senior level industrial automation course. This course is offered by distance (interactive video) to community colleges in the state. The industrial process consists of Human Machine Interface (HMI) software that utilizes a series of ActiveX controls to operate a Supervisory Control and Data Acquisition (SCADA/HMI) bottling process. The ActiveX controls were written by the author in Visual Studio. Each ActiveX control represents a component in the process, such as a switch, light, bottle, solenoid, cap, and conveyor. Individual components can be displayed and animated by writing code that connects them to the bottling process.
Students use the ActiveX controls to first program a simulation in Visual Studio 2010. They then connect to the bottling system using real time data from the process. Distance students can program Visual Studio to release a solenoid and allow the bottle to move to a fill station and monitor the system while a label and cap is placed on the bottle. The bottling process is controlled by a Remote Terminal Unit (RTU) that is connected to a central data server through MODBUS. Students access the central data server through a static IP address. This course gives students direct experience with modern industrial programmable controllers, wireless industrial networking, Radio Frequency Identification tags (RFID), and MODBUS.

Engineering Technology B.S. Program

The labs described in the course are part of a TAC/ABET accredited +2 B.S. Engineering Technology program that connects with ten community colleges. Students complete a two year engineering technology degree at the community college, then complete the bachelors primarily through live videoconferencing. One of the most challenging aspects of this program is the student laboratory experience. The focus of this paper is on a bottling process that allows for remote monitoring and control.

Comments on Engineering Laboratory Instruction

In the last thirty years there has been "major paradigm shift in technology, starting from analog to digital, macro to micro, from fixed (or wired) communication to mobile (or wireless) communication, etc."[1]. Tiwari also notes that there is a lag in traditional engineering laboratory experiences, especially with regards to remote monitoring and control. In the case of large equipment, the student is often reduced to spectator. The distance B.S. Engineering Technology program faces even more hurdles since students are in a location separate from the laboratory equipment.

Comments from Departmental Industry Advisory Council

For a number of years the author heard comments from the department's Industrial Advisory Council regarding topics that needed to be taught in industrial automation. Fifteen years ago it

was SCADA/HMI. Ten years ago it was Structured Query Language (SQL) for databases. Currently the comments revolve around remote monitoring and control of a process (at a recent meeting, one of the advisors showed a live robot in a clean room from Houston on his cellphone). All of these topics have been included in the Industrial Automation course. This course was first offered in Spring 2010. Wonderware was used to teach a local version of the class, but licensing issues prevented its use by distance students. Since the university has a site license for Labview that allows students to download it, consideration was given whether to use it. For example Northern Illinois University[2] is using Labview in a SCADA/HMI environment for an automation course. Also Labview can be fairly easily configured in a client/server environment.

In 2010, a survey was give to 200 graduates, members of the Industrial Advisory Council, and professionals who work in industrial automation. This survey asked what software and programming environments they used related to industrial automation. Six out of 200 respondents used Labview regularly in industry, 40% used Visual Studio in some form, and 85% used some sort of SCADA/HMI software. SCADA/HMI software included GE IFix, Allen Bradley RSView, Visual Studio, and Wonderware. Many also worked with a SQL database in a client/server network. All respondents said teaching of SCADA/HMI was critical to the course. Based on the results of this survey and Wonderware licensing issues, Visual Studio was chosen as an alternative to Wonderware and Labview.

Wireless and Wired Communication

Data is transferred to a computer from a process or machine using either a wired or wireless connection. Wired connections typically are Ethernet or serial. OPC, TCP/IP and Modbus protocols typically use this type of network[3]. Wireless connections normally use radio frequencies to transmit data. Examples include Bluetooth and RFID tags. In the process control industry, wireless communication is increasing in popularity, particularly in predictive maintenance. Here wireless "smart" sensors attached to machinery communicate with a server. The sensors automatically report when a degraded condition occurs on the machine.

Industrial Automation Course Prerequisites

Because of course prerequisites, all students have knowledge of circuit analysis, power electronics, programmable controllers and Visual Studio before starting the industrial automation class. They are required to purchase a CUBLOC 32M Industrial Remote Terminal Unit (RTU) Kit as part of the course. The CUBLOC can be programmed in BASIC or ladder logic and contains analog and digital I/O with relays. It also contains a MODBUS network for connection to other PLC/RTU and PCs. While low cost PLC units are available, the decision to use an RTU was based on improved system performance, communication flexibility, and configurability. Motorola states "In practice, the typical PLC usage model revolves around localized fast control of discrete variables. RTU usage focuses on remote monitoring with control, but with a higher demand for application communications and protocol flexibility[4]." A photo of the CUBLOC kit is shown in Figure 3:

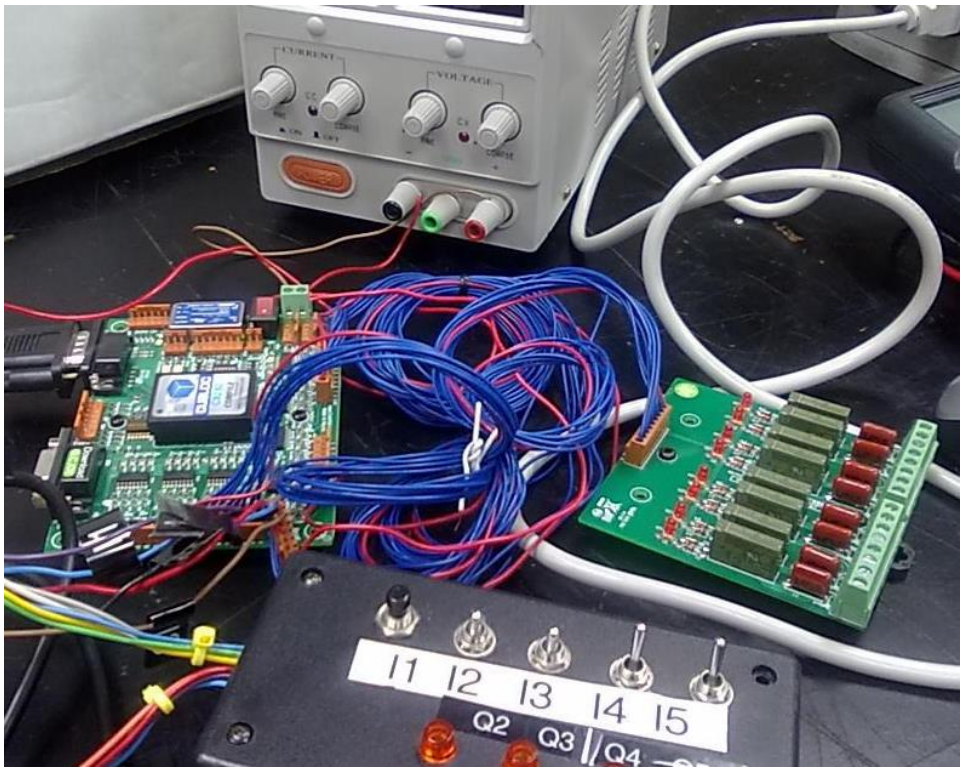

Figure 3: CUBLOC 32M with Switches

Students also purchase an inexpensive RFID reader with tags. The reader can be interfaced to Visual Studio through a serial port.

Bottling Process in the Industrial Automation Course

In 2009, a group of Senior Design students in the program began working on the bottling process. They designed and installed the fluid tanks, solenoids, fill station, label application, and cap feed. The label application proved difficult because it had to be applied at the bottle tipping point. The bottle would tip above or below this point. To find the tipping point, the center of gravity for various fill states had to be calculated.

The cap feed also proved difficult because the bottle tended to turn while the cap was being attached. After calculating the amount of torque needed to turn the cap, a device was designed to hold the bottle in place. The fill station and label feed are shown in Figure 1. Figure 2 shows the gripper of the cap attachment unit.

Figure 1: Bottling Fill Station and Label Feed

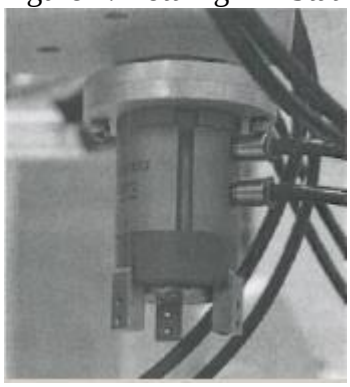

Figure 2: Cap Gripper

Industrial Automation Course Projects:

Currently students perform the following lab projects related to remote wireless monitoring and control as part of the Industrial Automation course.

- Program a TCP/IP client/server application in Visual Studio.
- Connect a MODBUS network from the CUBLOC to Visual Studio.
- Simulate a bottling process using Active X controls.
- Program an RFID reader.

- Write client/server code to remotely select a flavor and start an actual bottling process using the RFID reader.
- Write client/server code read in data and display a real time simulation of the bottling process.
- Connect to an SQL database and send data from the process to it.

The following is a description of each lab project:

Program a TCP/IP Client/Server
All students write code to create a local client/server connection on their computer. The connection between the client and server is shown in Figure 4.

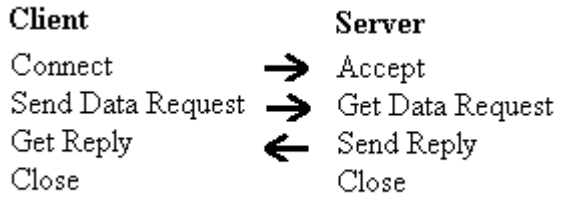

Figure 4: Client/Server Connection
Students practice sending data between a TCP/IP client and server utilizing network streaming and binary read/write commands. Sample code fragments given to the students are outlined in the appendix. The O'Brien text is a good reference on how to do this[5].

Connect a MODBUS Network
For this lab project, students first connect two CUBLOCs through a MODBUS network. They then have the switches on one RTU control the lights on the other. After successfully completing this, they write code to read the RTU data tables into Visual Studio using MODBUS.

Simulate a Bottling Process Using ActiveX Controls
SCADA/HMI is introduced through programmable ActiveX Controls in Visual Studio. These controls were written by the author. Students learn basic animation techniques in order to size and move the controls.

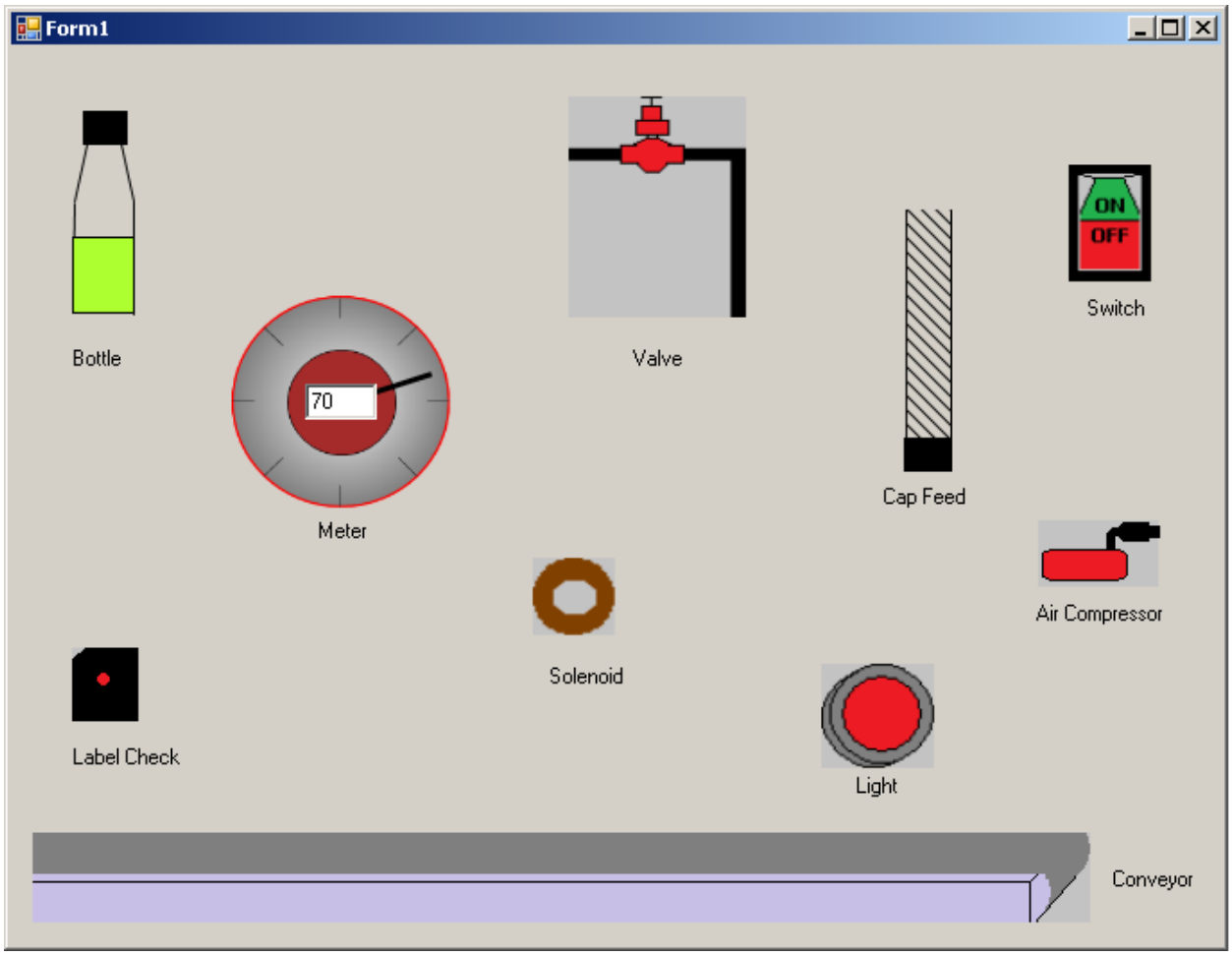

Figure 5: Visual Studio Active X Controls

For example the bottle control can be programmed to fill with various colors and have a cap placed on it. The meter can be scaled, and the light, valve, switch, label check, and air compressor can be turned on and off. This is done by setting variables in each control. Controls can be resized as needed.

Students load the controls onto the Visual Studio menu and activate them based on input received from switches on the screen. An example of the simulated HMI environment is shown in Figure 6:

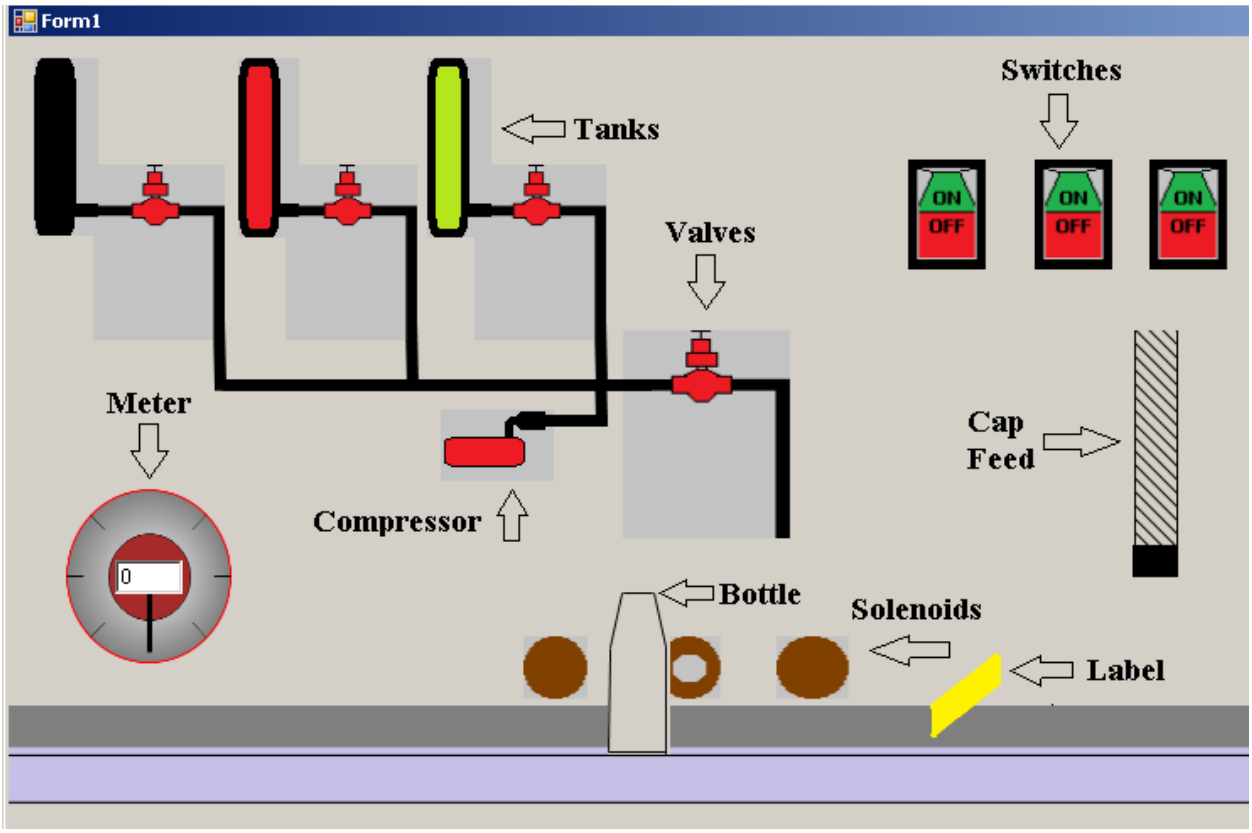

Figure 6: Simulated HMI Bottling Process

Sample code that sets the scale of the meter, determines the fill state of the bottle and turns a valve on is shown in the appendix.

Program an RFID Reader

RFID tags have become an important part of inventory management. Prior to 2011, barcodes were used in the industrial automation course to inventory the bottles in the process. Beginning in the Spring 2011, each student is now required to purchase and program an RFID reader as part of the course. The RFID is eventually used to select a flavor for the bottling process. The intent here is to help students become familiar with wireless ID tags. A sample screen showing an RFID tag reading is shown in Figure 7:

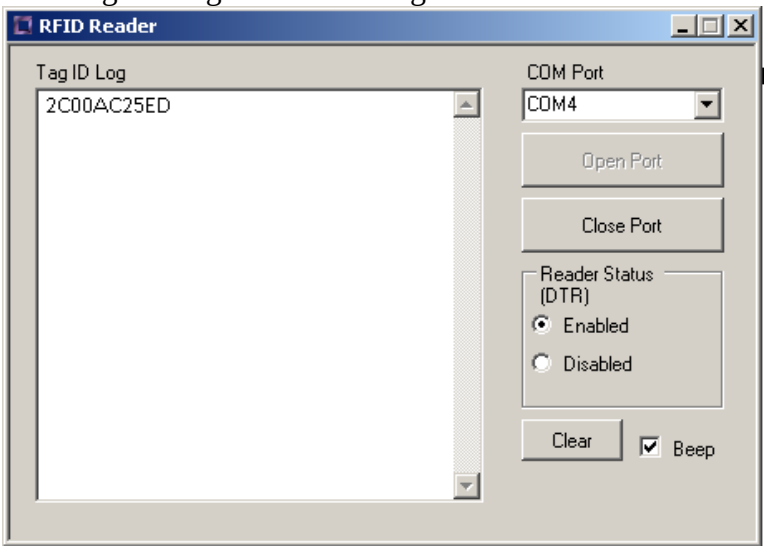

Figure 7: RFID Reader

Code to access a serial port for the RFID reader is shown in the appendix.

Connecting to CUBLOC

The connection to the simulated bottling process on a local client/server is shown in Figure 8.

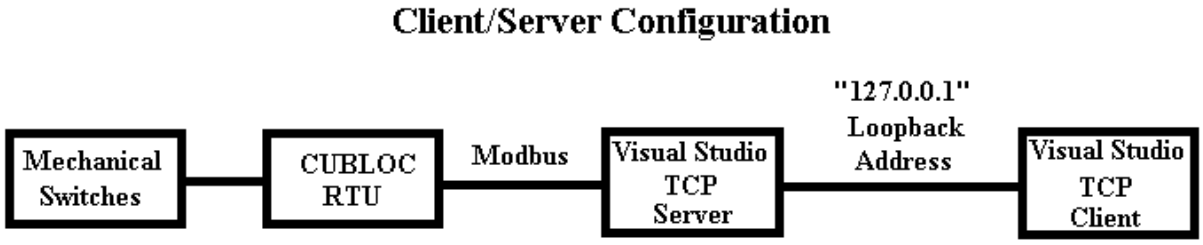

Figure 8: Bottling Client/Server Configuration

Students first write a CUBLOC BASIC routine to read in the switches and send their states over a MODBUS network to a PC. Visual Studio then sends the packet over a 127 local loopback address to a Visual Studio client programmed on the same machine. Students write all of the code for the CUBLOC and Visual Studio. The intent of this lab project is to help students prepare for a remote connection.

Write Code to Control a Real Time Remote Bottling Process

Next students write code to send data to a remote server. The server has a static IP address and is accessed through a VPN client. Students place ActiveX switches on the screen and pass the data to the remote server when the SEND button is pressed

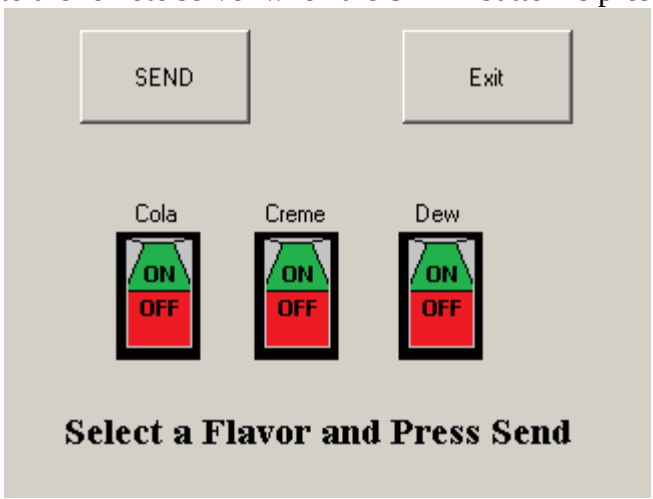

Figure 9: Flavor Selection Using Switches

Once the server receives a flavor, it sends a command through MODBUS to a CUBLOC that starts the process. After this lab project is completed, students revise the program to select the flavor using an RFID tag instead of a switch. The revised screen is shown in Figure 10:

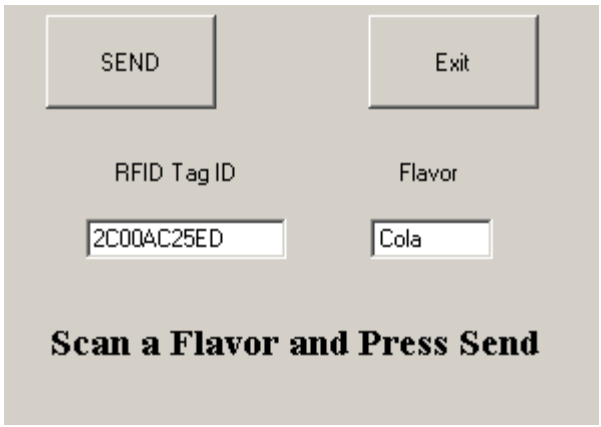

Figure 10: Flavor Selection Using RFID Tags

The data path from the RFID tag to the bottling process is shown in Figure 11:

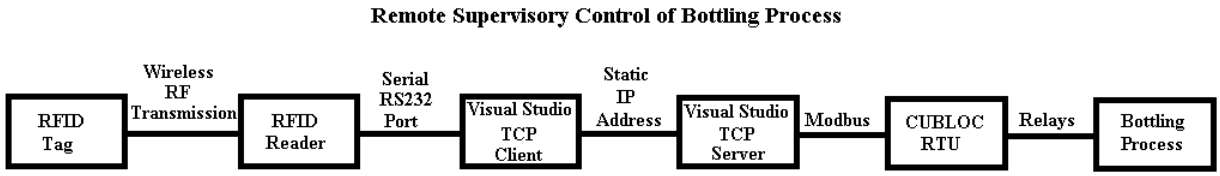

Figure 11: Remote Supervisory Control of a Bottling Process

Receive Data from the Remote Server and Display as an HMI

Next students write code to receive data from the remote server and use it to control the animation on their screen. Data is received as a MODBUS packet that must first be decoded. A sample HMI is shown in Figure 12:

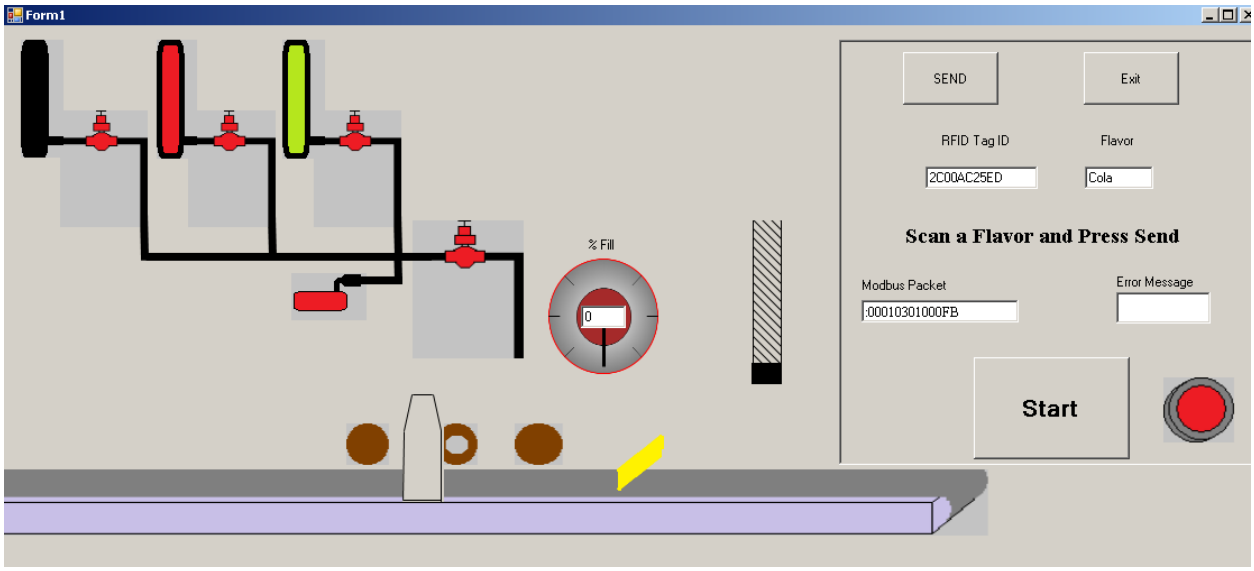

Figure 12: HMI Showing RFID Flavor Selection and Animation

The picture in Figure 13 was taken from a student's computer screen. It shows their HMI on the left, with the bottle being filled on the top panel and a label applied on the bottom panel. A video snapshot of the actual process is on the right.

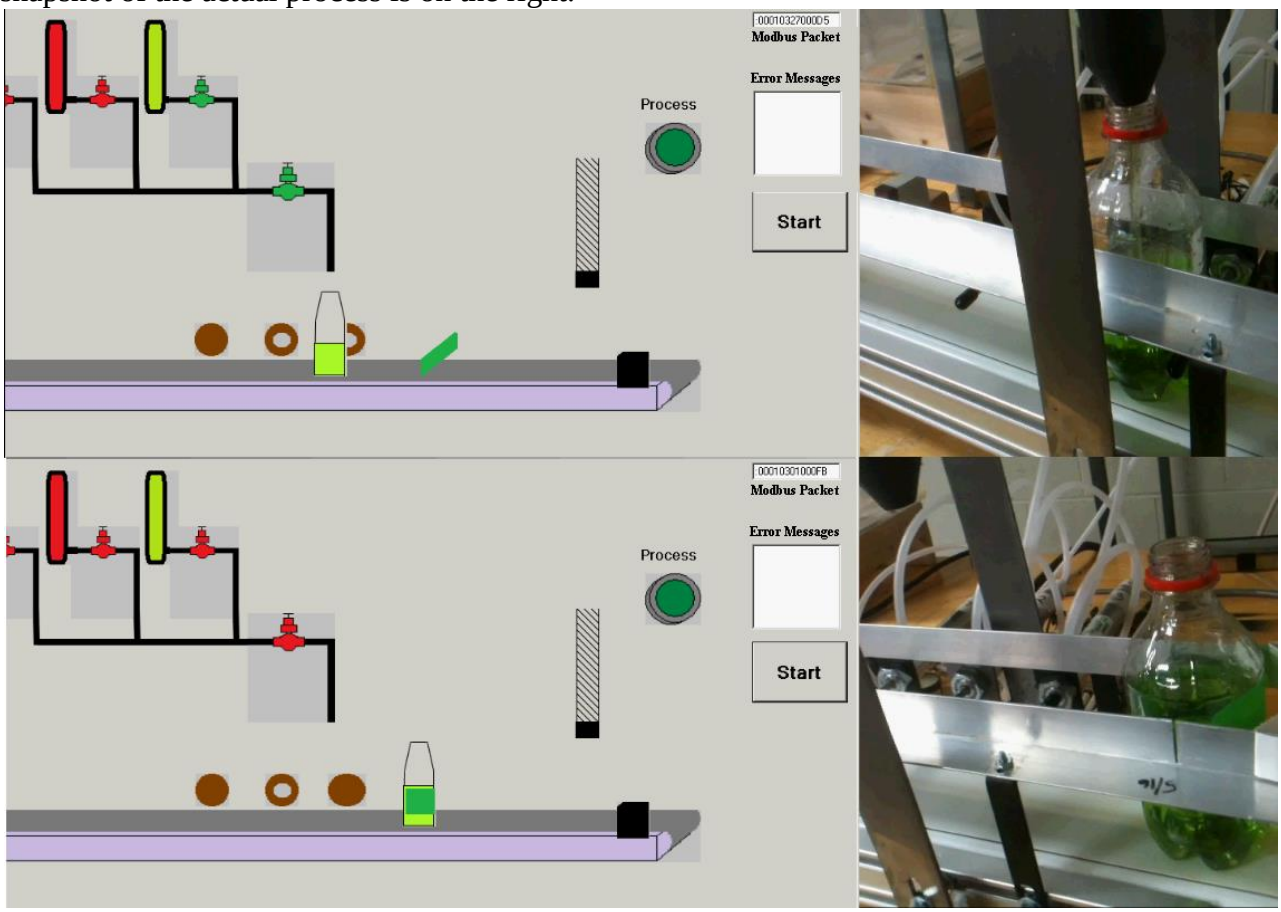

Figure 13: HMI and Video Screen

Interface to an SQL Database

After students complete the programming for the remote bottling process, they write code to connect and pass data to an SQL database. This is sometimes the most difficult part of the course. Flavors selected by the bottling process in a given day are passed by the server to the client. The flavors are then passed to SQL Management Studio through a Visual Studio BindingSource, Data Set, Table Adapter, and DataGridView. The Visual Studio template showing the DataGridView is shown in Figure 14:

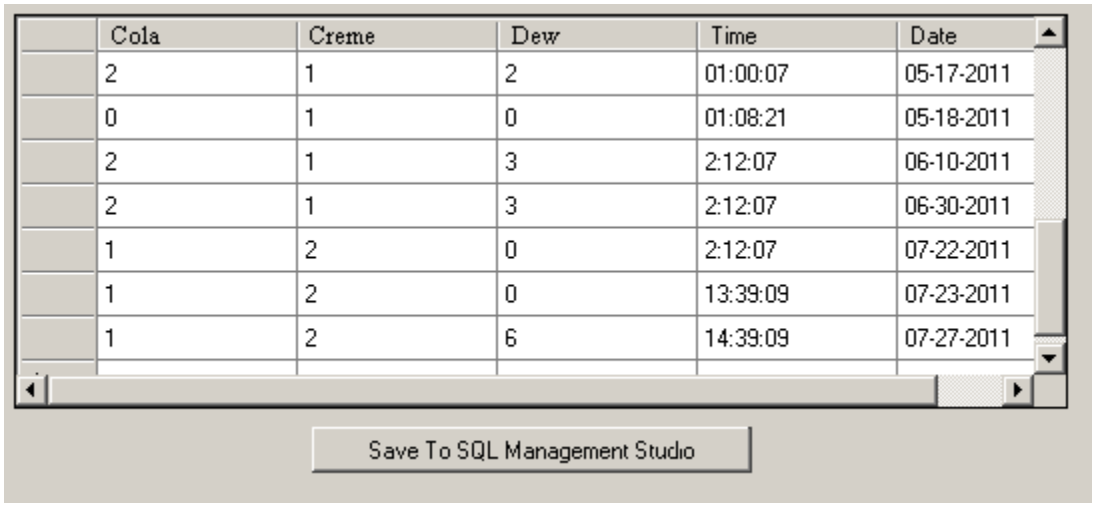

Figure 14: Visual Studio DataGridView

The SQL Management Studio showing sample data read from Visual Studio is shown in Figure 15, and the Binding Source and Table Adapter are shown in Figure 16.

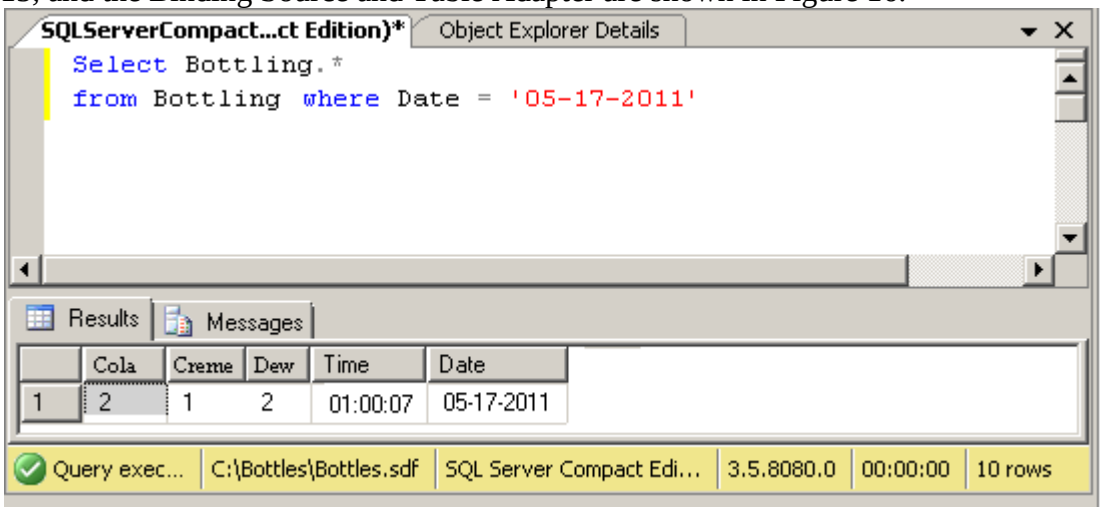

Figure 15: SQL Management Studio

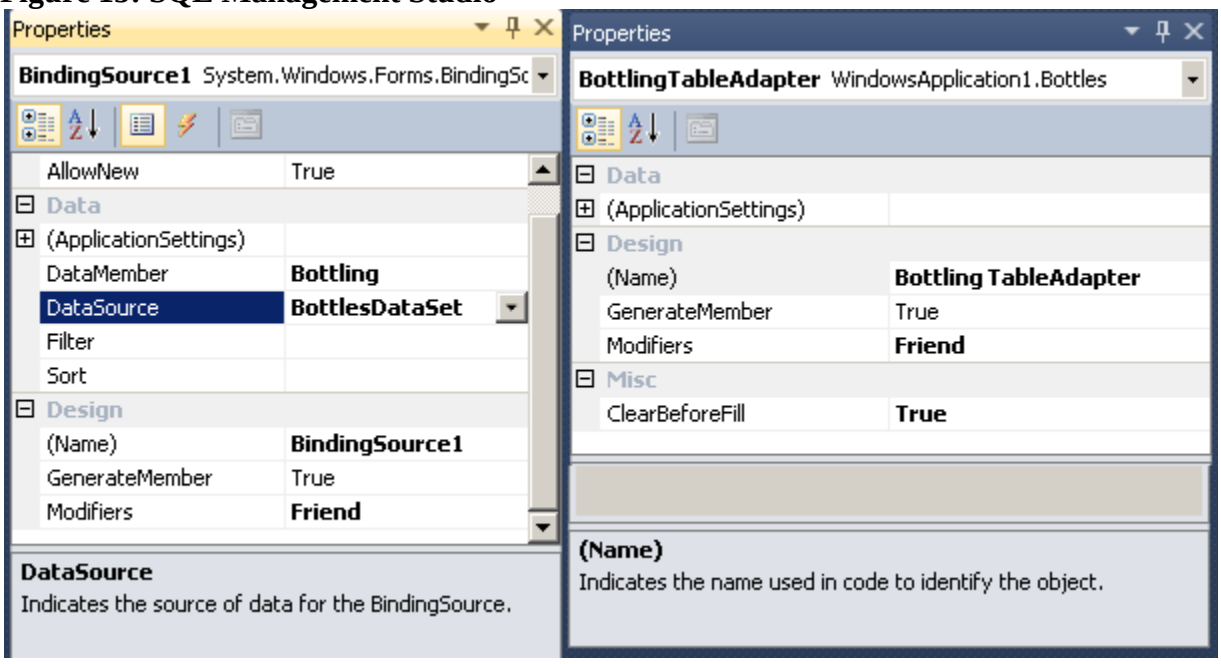

Figure 16: Visual Studio Binding Source and Table Adapter

Sample Visual Studio code for SQL connections is shown in the appendix.

Future Work

Senior Design students in the program are currently redesigning the bottling process to auto configure for various bottle sizes. It will then be used as a flexible automation demonstration in a Modern Manufacturing course.
A second group is working on bottle size and label recognition using a camera and image processing.
A third group is working on palletizing the bottles using a robot. It will adjust for various size bottles when packaging.
Finally the author is currently working on developing an iPad style app using HTML5 that would allow students to control and monitor the process.

Conclusions

While RFID tags are not a necessary component of a bottling process, the intent is to give students at the distance sites some exposure to wireless communication. In industry, the popularity of RFID tags has rapidly increased. In 2005 it was predicted that they would soon replace barcodes for inventory applications[6]. However at present this has not been the case. Communication problems (including noise and ground plan interference[7]) are the primary reason their use has leveled off. Still, RFID tags are increasing being used as a method of inventory tracking and identification.
Feedback from the departmental Industrial Advisory Council on the bottling process labs has been very positive.

Bibliography

1. Tiwari, R., Singh, Khilawan. "Virtualisation of Engineering Discipline Experiments For an Internet-Based Remote Laboratory", Australasian Journal of Educational Technology, pp 671-692

2. Otieno, A. "Development of SCADA Experimental Systems Through Student Projects To Enhance the Automation Curriculum in a Manufacturing Engineering Technology Program", ASEE Annual Conference Proceedings 2007

3. http://www.reliabilityweb.com/articles/remote_machinery_condition_monitoring.pdf

4. http://www.motorola.com/web/Business/Products/SCADA%20Products/_Documents/Static%20Files/SCADA_Sys_Wht_Ppr-2a_New.pdf

5. O'Brien B, Seaver C, Microsoft Visual Basic.NET Programming Essentials ,pp 290-353 Microsoft, 2004

6. http://www.cioupdate.com/trends/article.php/3550656/RFID-Gaining-in-Popularity.htm

7. P. Raumonen, L. Sydanheimo, L. Ukkonen, M. Keskilammi, M. Kivikoski, "Folded dipole antenna near metal plate," Antennas and Propagation Society International Symposium, 2003.

Appendix:

Sample RS232 serial port commands:

```
'Declare com1 as serial port
 Dim com1 As IO.Ports.SerialPort = Nothing

'Set to COM4
 com1 = My.Computer.Ports.OpenSerialPort("COM4")

'Set to 9600, no parity, 8 bits, 1 stop bit
 com1.BaudRate = 9600
 com1.Parity = IO.Ports.Parity.None
 com1.DataBits = 8
 com1.StopBits = 1

‘Set Handshaking
com1.DtrEnable = False
com1.RtsEnable = False

'Time out after 10 seconds
com1.ReadTimeout = 10000

If com1 IsNot Nothing Then
'Read byte from COM4
Reading = Val(com1.ReadByte())

'Place reading in text box
TextBox2.Text = Reading

'Close connection
com1.Close()

End If
```

Sample ActiveX Commands

```
'Turn Valve 2 on
Valve2.State.Text = "On"

'Select Brand
BottleFill1.Brand.Text = "Cola"

'Set Fill Level to 50%
BottleFill1.FillLevel.Text = 50

‘Move Bottle to left 10 pixels
BottleFill1.Left= BottleFill1.Left+10

'Display Label
```

```
 BottleFill1.Label.Text = "On"

'Set minimum scale on meter
Meter1.Minimum.Text=0

'Set maximum scale on meter
Meter1.Maximum.Text=100

'Set scale reading to 70
Meter1.Reading.Text=70
```

Sample TCP/IP Client Connection

```
'Dim Client
 Dim Client As New TcpClient

'Connect to Loopback Address 127.0.0.1, port 8000
Client.Connect(("127.0.0.1"), 8000)

 'Dimension network stream and binary read/write
 Dim Stream As NetworkStream = Client.GetStream()
 Dim r As New BinaryReader(Stream)
 Dim w As New BinaryWriter(Stream)

 'Request a connection from server
  w.Write(ClientMessages.RequestConnect)

'Wait for response from server
 If r.ReadString() = ServerMessages.AcknowledgeOK Then

'Send contents of Textbox1
 w.Write(TextBox1.Text)

'Disconnect from server
 w.Write(ClientMessages.Disconnect)
 End If

'Close network connection
Client.Close()
```

Sample Code To Configure Visual Studio for SQL Connection

```
    Dim tableadapter As New BottlesDataSetTableAdapters.BottlingErrorsTableAdapter


        tableadapter.Insert(TextBox1.Text)
        tableadapter.Fill(BottlesDataSet.BottlingErrors)
        tableadapter.Update(BottlesDataSet.BottlingErrors)
```