Integration of Enterprise and Industrial Networks in Computer Engineering Technology Program


Purdue School of Engineering & Technology
Indiana University-Purdue University at Indianapolis, Indiana
*Horner APG
Indianapolis, Indiana

Abstract

In many industrial plants, the local area network is a relatively small path that connects computers and workstations used by managers and engineers. Unlike enterprise networks, industrial networks are typically dedicated to conveying critical control information and operational data to operators, equipment, controllers, valves, and sensors. Due to this nature, normal industrial networks are usually kept isolated from the enterprise networks. In the same manner, the general approach to curricular structure designs in enterprise networks and industrial networks is to separate the two. However, the rapid growth of information technology and the continuing cost reduction in computing hardware have stimulated the growth of computer networking in all aspects. The interoperability between these two types of networking becomes an important and valid issue to be addressed. Vendors and developers of industrial and enterprise networks are trending toward integrating these two types of networking with designs of interoperable protocols and proper network architectures.

In this paper, we discuss the impact of this trend on the design of curriculum in the Computer Engineering Technology program. We also report our initial attempt in assimilating these two networks from the curriculum point of view. The related course material and associated laboratory exercises used in this initial attempt and their implementation in the higher-level curriculum in Computer Engineering Technology program are discussed.
I. Introduction

The ultimate goal of collegiate education in engineering and technology is to help students mature into skillful and responsible problem-solvers. For years, potential employers of our graduates have expected their new college hires to be technologically conscious and job-ready as well as being capable and efficient problem-solvers. The exponential growth in the telecommunications industry and information technology has created a great demand for skilled graduates who can quickly adapt into this continually-changing technological industry. This demand quickly inspired similar growth in the curricular design supporting the requirements of this changing industry throughout all levels of education. Today’s corporations and industry are increasingly dependent on educational institutes to prepare their graduates in readiness and awareness of the fast growing technology.

The Internet took a major leap in the 90s, and it is now used by corporations and industry as a means to reach their employee as well as customers. With the increasing use of Internet as a conduit to share information between entities within the organization, and as a medium to transport information and services to customers through World-Wide Web and Internet applications, corporations and industry are increasingly dependent on their LAN and WAN as information resources for daily routines and revenue generation. Unlike the Internet and enterprise network, which is built for large clients accessing a relative small number of data servers, industrial networks are typically dedicated to conveying critical control information and operational data to operators, equipment, controllers, valves, and sensors. Industrial network infrastructure also offers its content to an audience, but often to a small one. Frequently, small controllers operate the industrial infrastructure. Due to its nature, normal industrial networks are usually kept isolated from the enterprise network. In the same manner, the general approach to curricular designs in enterprise networks and industrial networks is to deal with them separately. However, the rapid growth of information technology and the continuing cost reduction in computing and switching hardware have stimulated the growth of computer networking in all aspects. The synergy between these two types of networks seems inevitable. The interoperability between these two types of networks becomes an important and valid issue to be addressed. In fact, vendors and developers of industrial and enterprise networks are trending toward integrating these two types of networks with designs of interoperable protocols and proper network architecture.14

Control systems are continuing to become more complex. They require networks to reduce labor and equipment cost, and to support the continued growth. These more complex networks are producing valuable data that users are requiring on their enterprise networks. Many controls engineers are struggling with these rapid changes in the industry. This curriculum attempts to provide the background and experience to get started in today’s networked control systems.

II. Background and the impact on the curriculum
Typical upper-level computer networking curricula in technology programs include courses in data communications, networking technologies, and a sequence of computer networking courses, such as LAN, WAN, and enterprise network. Generally, these courses evolve from traditional electronics communications sequence with the influence of Internet architecture. The curricular designs are more or less oriented around the Internet protocol model, TCP/IP protocol suites. On the other hand, the curricula in industrial controls and automation are exemplified by courses like introduction to control systems, and automation, instrumentation, and process control, etc. The industrial networking issues are generally covered at the upper-level courses as a brief exposure to students. The integration and the assimilation of course material of these two sequences are seldom to be seen.

At IUPUI, our Computer Engineering Technology (CpET) program was approved to be offered by the Department of Electrical and Computer Engineering Technology (ECET) in 2001. The CpET program has two options available, namely, the industrial computing option and the telecommunications option. The industrial computing option is designed to facilitate and to nurture students into mature, skillful problem solvers in industrial computing environment. With a traditional Purdue engineering technology program, the curriculum in Electrical Engineering Technology (EET) was already rich in industrial controls area; the emphasis in this option is to prepare students more in computer hardware/software, computer networking and data communications aspects. Curriculum in this option includes: computer hardware courses, starting from microprocessors to PC systems; software programming courses, ranging from low-level to high-level programming languages; computer communications, from data communications to various computer networking environments. Students are expected to be equipped with strong background in both industrial controls and information technology areas.

Even though during the designing stage of this new program, the integration trend of computer networking in industrial environment has been envisioned and the curricular design has followed the trend accordingly, the delivery of the course material has not yet been able to reflect as it was originally envisioned. As an example, a typical CpET student in the industrial computing option is required to take ECET 231, Electrical Power & Controls, as the first course in industrial controls, followed by ECET 371, a course in automation, instrumentation, and process control for their industrial concentration. In these two courses, the student is exposed to subjects such as control devices, programmable logic controllers, PLC input and output devices, PLC communications, automatic testing, computer interfacing, data collection, robotic controls, and graphical process control software. In the computer networking sequence, the same student is required to have ECET 284, Computer Communications, and ECET 483, Networking Fundamentals with Microcontrollers. In these two courses, the student learns the concepts in data communications, networking technologies, computer networking, protocols, system software, and eventually focuses on the usage of data communication techniques and their applications in the industrial environment. However, a practical limit in teaching resource, lack of an instructor knowledgeable in both areas, restricts the action of assimilating material between these two areas. Based on students’ feedback in course evaluations through related courses, it seems that the conceptual integration
between enterprise network and industrial network does not exist for those students who have finished these courses. In order to address this issue, an attempt was initiated in the fall semester of 2003 to include a two-week guest lecture and laboratory exercises on the subject of industrial networks in the upper-level class, ECET 483. An individual with expertise in industrial networks was invited from the industry sector to deliver the lectures and to supervise the laboratory exercises. In addition, the lecture portion was also opened to students in a lower-level class, ECET 231, who already had the necessary concept of PLC communications by the time of lecturing. The responses through immediate feedbacks from both sets of student right after the lecture periods were quite encouraging. Most of the responses from upper class students confirm the particular course objective related to the usage of data communication techniques and their applications in the industrial environment. On the other hand, although these lectures presented themselves as a series of seminars to students in the lower-level class, their responses were positive and confident that inspired their anticipation for the coming computer networking course. As course objectives of related courses in these two areas are continuously being refined to reflect the integration trend, more assessment data can be collected and reported in the future.

In the following section, we illustrate the principles of selecting lecture topics and laboratory exercises in this two-week period in an attempt to assimilating these two different kinds of network. The detailed outlines of this two-week guest lecture and the two laboratory exercises are provided in the Appendix.

III. Selection of course material and laboratory exercises

The goal of inserting guest lectures and laboratory exercises in the subject of industrial networks is to provide our students in the CpET program a comprehensive understanding on the current trend of integration and convergence between industrial networks and enterprise networks. As this is the first time that industrial networks are integrated in an upper-level computer networking course in our CpET program, we also took into account the backgrounds provided by early courses in the program for both levels of class. Since the existing subjects covered in both industrial controls and computer networking areas in the program are relatively sufficient in depth, our emphasis on this two-week addition concentrates more on the awareness of the trend and pathways that bridge the two different networks.

A general introduction on industrial network is given at the beginning of the lectures. This introduction provides an overview of industrial networks for students in the upper-level class and brings up the common concepts existed between enterprise networks and industrial networks. Meanwhile, this introduction serves as an opening for students in the lower-level class so that the continuity of course subjects in the lower-level class can be observed. The network-dependent layers such as physical interfaces and data link layer are then brought up to the students’ attention and described in a more detailed manner. The possible pathways that bridge the two different networks are exemplified by description and examination of existing protocols such as OPC data interfaces. Various higher-layer protocols in industrial Ethernet (iE) are evaluated during the latter half of the
The lecture period. The lecture series ended with discussions on issues unique to industrial networks as well as common issues on both types of network. It is our hope that students will become aware of the trend of integration and convergence between these two networks and inspire themselves to further engage in enhancing their knowledge in a more comprehensive manner.

The detailed outlines of this two-week guest lecture and the two laboratory exercises are provided in the Appendix.

IV. Challenges and conclusion

An attempt to cover ongoing computer networking issues in an existing curriculum is always a challenging task. While members in the industry are still debating issues related to networking and slowly realign themselves in a proper direction, academic curricula need to be designed with this future industry in mind. The rapid growth of the information technology increases the complexity of today’s industry even more. Traditional disciplinary curricular design approach may not provide enough of the type of experiences students will encounter in the industry. Seeking an ingenious approach in curricular design seems to be a never-ending process. In this paper, we report our initial attempt to assimilate the two different network infrastructures from the curriculum aspect. The addition of lectures and labs in industrial networks into a computer networking course does not necessarily address the integration issue completely. However, it is an initial step in better preparing students in their undergraduate Computer Engineering Technology program for careers in industry, business, and commerce.

V. Appendix

Course outlines for industrial networks

What is the purpose of an Industrial Network?

- Distributed I/O
  - Mount I/O on the machine to simplify the wiring
- Distributed Control
  - Parallel processing – many CPUs to do one large task
  - Global Data
    - Peer to Peer
- Master Slave
- Controller Supervisory
  - Download, upload, monitor, debug
- Data Logging
- HMI – Human – Machine Interface
  - PC or Embedded solution to display Data / Accept User inputs

General Physical Interfaces

- Wiring
- Characteristic Impedance
- Capacitance
- Wire Gauge
Shielding

Networks with Power
DeviceNet Example
Determining Power requirements
Placement of Power supplies
Termination
Use resistors based on cables characteristic impedance
Prevent reflections
Connectors
Grounding
Wiring Topology
Isolation
Length Limits
CAN
Ethernet
RS-485
Using repeaters
Active
Passive

Serial Communications
Simple Shift Registers
UARTS
Communications ASICs / Peripherals

Physical Standards
RS-232
RS-485
CAN
Profibus
Ethernet

Example Industrial Networks
Modbus
Master Slave Network
ASCII and RTU formats
Usually 9600 baud RS-485
247 nodes per network
CsCAN
Uses CAN physical layer
Peer to peer data
253 nodes per network
125k, 250k, 500k, 1meg CAN
Global Data
Change of State
Change of Value
Periodic Transmission
Controller initiated transmission
Heartbeats
Host to Node
Pass through communications
DeviceNet
Uses CAN physical layer
Master-Slave with peer to peer
64 nodes per network
125k, 250k, 500k CAN
Polled Data
Change of State Data
Bit Strobe
CAN Open
Uses CAN physical layer
125k, 250k, 500k, 1meg CAN
Master Slave with peer to peer
Profibus
Uses custom differential pair signaling
Master-Slave
32 nodes per network
Up to 12 Meg

**Network Performance Characteristics**
- Network latency
  - time from sender to receiver
- Turn around time
  - time receiver takes to respond
- Bandwidth
  - amount of data the network can carry (KB/sec…)

**Network Trouble Shooting**
- Bandwidth limitations
- Excessive collisions
- Electrical noise
- Poor connections
- Viewing differential signals on an Oscilloscope

**Interfacing PLCs with PC software**
- Proprietary Software Interfaces
  - Programming tools
  - Configuration packages
  - Often simpler, faster and more efficient
- OPC data interfaces
  - Industry standard
  - OPC servers usually provide OPC and DDE interfaces
  - DDE for WORD, EXCELL, ACCESS
  - OPC used for HMI packages

**Wireless Networks**
- Radio modems
- Cellular modems
- Wireless Ethernet (802.11)
- Wireless gateways (CAN, serial…)
- Low-powered radios

**Fiber Optics**
- Ethernet, CAN
- Plastic vs. Glass

**Using redundant networks**
- Having backup data paths if one fails

"Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition
Copyright © 2004, American Society for Engineering Education"
Ethernet Industrial Networks

Ethernet & TCP/IP
Brief review of Ethernet and TCP/IP

Example Protocols
Modbus TCP
Very similar to serial RTU Modbus, packets transferred in TCP packet
Very good for HMI or Master / Slave distributed control
SRTP (GE)
TCP version of the proprietary SNP serial protocol
Very good for HMI or Master / Slave distributed control
Used for PLC supervisory functions
EGD (GE)
Uses UDP data packets
Data is sent periodically
Receiver has timeout to expect data
Up to 1400 bytes per packet
Configuration Demo
CsCAN TCP (Horner)
TCP version of the CsCAN Host to Node protocol
Used for PLC supervisory functions
Security

SMTP
Controllers / HMI send email on alarm or notification event
HTTP
Controller / HMI provide a web interface to view / interact with the control process
FTP
Collect data
Load new program / interface

Administration / Infrastructure
Hubs
Switches
Routers
Firewalls

Ethernet Timing
PC Stack / Controller Latency
Switch Latency
Data collisions
Lost / Damaged Packets

Network Security
Firewalls
Private Networks
Gateways
Using vendor supplied tools for security

Choosing an Industrial Network
Cost
Hardware
Software
Labor
Laboratory Exercises

Setting up and Using Industrial Networks

OBJECTIVE:
1) To wire, troubleshoot and use a peer to peer industrial network.
2) To calculate the theoretical and actual bandwidth usage for different loads on the network.
3) To calculate the maximum network throughput.
4) To configure a typical master-slave industrial network.

EQUIPMENT AND MATERIALS:
Workstation capable of running MS Windows 95 or newer.
Cscape Programming Software 6.10 or newer.
OCS – controller with operator interface and integrated CsCAN network
Power supply and programming cable
SmartStix CsCAN based remote I/O
Wire approved for CAN network and two terminating resistors
Oscilloscope

PROCEDURE:
Wiring the network
1. Using one of the twisted pairs in the cable, connect the CAN_H from the OCS to the CAN_H on the Remote I/O. With the other wire in the pair, connect the CAN_L from the OCS to the CAN_L on the remote I/O.
2. Connect one 120Ω resistor across the CAN_H and CAN_L terminals at the OCS. Connect a second 120Ω resistor across the CAN_H and CAN_L terminals at the remote I/O side.
3. Using a different pair, select a wire and connect the V+ from the OCS to the V+ on the remote I/O. At the terminal connected to the OCS, connect the V+ to a 24 Volt DC power source. Hint: The OCS uses 24 VDC as a power source and the power connect is inches away from the CAN connector.

---

“Proceedings of the 2004 American Society for Engineering Education Annual Conference &Exposition
Copyright © 2004, American Society for Engineering Education”
4. If your networking cable has a drain wire (non-insulated wire), connect it to the shield terminal ONLY on the OCS side.

5. Using a different pair than the one from step one, select a wire and connect the V+ from the OCS to the V+ on the remote I/O. At the terminal connected to the OCS, connect the V+ to a 24 Volt DC power source. Hint: The OCS uses 24 VDC as a power source and the power connect is inches away from the CAN connector.

6. Use any remaining wires to connect the V- from the OCS to V- on the Remote I/O. At the OCS side connect the V- to the same 24 volt power supply used in the previous step.

7. Apply power to the OCS. Make sure the power LED on the remote I/O illuminates.

**Testing the SmartStix I/O**

1. Open the Cscape application.

2. Set the network ID of the OCS to 1 by using the menu item Controller|Set Local Network ID.

3. Set the network ID of the remote ID to 2 by moving the rotary switches on the top of the unit.

4. Enter the following ladder program:

![Ladder Diagram](image)

5. Download the program to the OCS using the menu item Program | Download.

6. Make sure the OCS is in run mode by selecting Controller Run / Monitor.

7. Press the F1 key on the OCS and see if the first digital output on the remote I/O turns ON.

**Determine the theoretical network bandwidth**

Every time the data changes, approximately 63 bits of information are transmitted. The CAN network is running at 125,000 bits per second.

1. If the data changes 20 times per second, what percentage of the available bandwidth is used?

2. If the data changes 200 times per second, what percentage of the available bandwidth is used?

**Find the Actual network bandwidth**

1. Modify the above program to match the example below:

![Modified Ladder Diagram](image)

"Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition Copyright © 2004, American Society for Engineering Education"
2. Download the program. This will change the network I/O 20 times per second (on and off every 100 mS).

3. Find the actual network usage by selecting Controller | Status from the menu and look at the “Target Net Usage” item.

4. Now change the contact from %S4 to %S3 and download the program. This will change the network I/O 200 times per second (on and off every 10 mS).

5. Find the actual network usage for this program.

Calculate the maximum CAN network throughput

In this configuration each network message requires 63 bits, the network allows 125,000 bits per second.

1. How many messages per second is maximum allowed?

2. In this configuration each message carries 16 bits of actual I/O data. Based on this what is the maximum throughput of this network in bits per second?

3. Recalculate this for a “full” CAN packet that requires 111 bits of data and carries 64 bits of actual I/O data.

View the differential CAN signals using an oscilloscope

1. Connect the CAN_H signal to channel one of the scope. Hint: the resistor makes a great place to clip the scope leads. Make sure this signal is viewable, triggered and is scaled to take slightly less than one third of the display.

2. Connect the CAN_L signal to channel one of the scope. Make sure this signal is viewable, triggered and is scaled to take slightly less than one third of the display.

3. As discussed in class use the digitals scopes math function to subtract channel 2 from channel 1.

4. Stop the scope and document all three signals.

Configure a DeviceNet network

For this portion of the lab, you are to configure a DeviceNet network with a master and two slaves. Because software for configuring the networks can vary from vendor to vendor and from product to product, the step by step directions have been left off. The goal is to become familiar with the software so you can add two slave devices as specified below.

1. Start the Cscape software.

2. Select Controller | IO Configure from the menu.

3. Find an empty I/O slot and insert a DNT450 Devicenet Master I/O module (under the Comm tab).

4. Click the “Config” button beside the button, select the “Module Setup” tab, and click the “Configure” button.

5. For the initial dialog all the default parameters are acceptable.

6. Configure a slave with the following parameters:

   Input Data: 12 Words
   Output Data: 7 words
   Expected packet Rate: 2000 mSec
   We want to update this data every 200 mSec.

7. Configure a second slave with the following parameters:

   Input Data: 32 bits
   Output Data: 16 bits
   Expected packet Rate: 500 mSec
   We want to update this data every 50 mSec.
Using OPC Servers and Ethernet Networks

OBJECTIVE:
1) Setup and use an OPC server with a DDE client to display and plot dynamic data in Excel.
2) Setup and use a Modbus Ethernet (master-slave) network to exchange data with a controller.
3) Setup and use a GE EGD (peer-to-peer) network to exchange data with a controller and other PCs.
4) Observer different network topologies.

EQUIPMENT AND MATERIALS:
Workstation capable of running MS Windows 2000 or newer.
Microsoft Excel (Office 2000 or newer)
Lab CD Containing:
- Cscape Programming Software 6.10 or newer
- CsCAN OPC Server Lite 1.3 or newer
- Kepware KepServerEx 4.102 (demo mode)
- OPC_Example.csp (Cscape program file)
- OCS100 – controller with operator interface and integrated CsCAN network with power supply and programming cable
- SmartStix CsCAN based remote I/O
- Wire approved for CAN network and two terminating resistors
- One shared OCS451 (Ethernet enabled) connected to the local Ethernet network

PROCEDURE:

Using an OPC server with a DDE Application (Excel)

1. Make sure your PC is setup on a local network with an IP address of 192.168.1.X, where X is a number from 1 to 100. Make sure you select an IP address different from the other PCs on your network.
2. Install the Cscape programming software if you haven’t already done so. (This can be found on the Cscape directory on the lab CD.
3. Install the CsCAN OPC Lite software.
4. Connect the serial of the OCS100 to the serial port of the PC. Start the Cscape software and open the file “OPC_Example.csp”. Download the file to the OCS by selecting Program | Download. Make sure the OCS is in RUN mode and is set to network ID “1”.
5. Close Cscape and start the CsCAN OPC server (located in your Start | Programs menu).
6. Open Excel. In Cell A1 enter: =Cslite|Net1_Node1!'R100'. In cells A2 to A10 enter =Cslite|Net1_Node1!'R101' to =Cslite|Net1_Node1!'R109'. This should be displaying 10 numbers that are periodically changing.
7. Use Excel to plot the values in A1 to A10.
Using an OPC server/client with a Master-Slave Ethernet Industrial Network

For this portion of the lab your PC will be communicating with a Ethernet enabled OCS (PLC) with an IP address of 192.168.1.200. You should start by pinging this address to make sure communications are function properly.

1. Install the KepWare Server application. Make sure to install the KEPServerEx with the Modbus Ethernet drivers, the GE Ethernet Global Data and the OPC Quick Client. No other options are required for this test.

2. All the groups will wire their SmartStix remote I/O device to a single CAN network. This network will connect to the Ethernet enabled controller. Each group will be assigned a CAN ID from 2 to 9. Set your SmartStix unit to the assigned ID using the rotary switches. A ladder program is running in the OCS that maps the remote I/O as follows:

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>%I1-%I16</td>
<td>%Q1-%Q16</td>
</tr>
<tr>
<td>3</td>
<td>%I17-%I32</td>
<td>%Q17-%Q32</td>
</tr>
<tr>
<td>4</td>
<td>%I33-%I48</td>
<td>%Q33-%Q48</td>
</tr>
<tr>
<td>5</td>
<td>%I49-%I64</td>
<td>%Q49-%Q64</td>
</tr>
<tr>
<td>6</td>
<td>%I65-%I80</td>
<td>%Q65-%Q80</td>
</tr>
<tr>
<td>7</td>
<td>%I81-%I96</td>
<td>%Q81-%Q96</td>
</tr>
<tr>
<td>8</td>
<td>%I97-%I112</td>
<td>%Q97-%Q112</td>
</tr>
<tr>
<td>9</td>
<td>%I113-%I128</td>
<td>%Q113-%Q128</td>
</tr>
</tbody>
</table>

“Proceedings of the 2004 American Society for Engineering Education Annual Conference &Exposition
Copyright © 2004, American Society for Engineering Education”
3. Start the KEPServerEx software package and create a new project (File | New). Add a new channel. Use all default selections except the device driver (select Modbus Ethernet).

4. Add a device to the network. Use all the default except the device address that should be set to 192.168.1.200.

5. The chart below shows how the registers in the OCS are mapped into standard Modbus address. Create four tags two for the first two inputs assigned to your groups remote I/O and two for the outputs. Make sure to make the inputs read only, and the outputs read/write.

<table>
<thead>
<tr>
<th>ControllerReference</th>
<th>Max</th>
<th>Traditional ModbusRef</th>
</tr>
</thead>
<tbody>
<tr>
<td>%I1</td>
<td>2048</td>
<td>10001</td>
</tr>
<tr>
<td>%IG1</td>
<td>256</td>
<td>13001</td>
</tr>
<tr>
<td>%S1</td>
<td>256</td>
<td>14001</td>
</tr>
<tr>
<td>%K1</td>
<td>256</td>
<td>15001</td>
</tr>
<tr>
<td>%Q1</td>
<td>2048</td>
<td>1</td>
</tr>
<tr>
<td>%M1</td>
<td>2048</td>
<td>3001</td>
</tr>
<tr>
<td>%T1</td>
<td>2048</td>
<td>6001</td>
</tr>
<tr>
<td>%QQ1</td>
<td>256</td>
<td>9001</td>
</tr>
<tr>
<td>%AI1</td>
<td>512</td>
<td>30001</td>
</tr>
<tr>
<td>%AIG1</td>
<td>32</td>
<td>33001</td>
</tr>
<tr>
<td>%SR1</td>
<td>32</td>
<td>34001</td>
</tr>
<tr>
<td>%AQ1</td>
<td>512</td>
<td>40001</td>
</tr>
<tr>
<td>%R1</td>
<td>2048</td>
<td>43001</td>
</tr>
<tr>
<td>%AQG1</td>
<td>32</td>
<td>46001</td>
</tr>
</tbody>
</table>

6. Start the OPC client by selecting Tools | Launch OPC Quick Client. Apply 24 volts (available at the network connection) to the input common and the first and second inputs. Change the first output by right clicking and selecting “Synchronous Write”. Record the results of the various stimuli.
7. Observe and describe the network topology of this lab setup. Make sure to include your fellow lab groups’ PCs and the remote I/O devices in the description.

Using an OPC server/client with a Master-Slave Ethernet Industrial Network

1. With the KEPServerEx software still open, create a new project by selecting File | New. Create a new channel, again using the default except selecting “GE Ethernet Global Data” as the device driver.
2. Add a new device. Use all the defaults we will add network information later.
3. Right click on the exchange and select “Properties” then select the “Exchange Configuration” tab. Add a consumed exchange with the following information:

This will consume data from the OCS being broadcast to the group with a group number of 20 and an exchange number of 200.
4. Add a range of %R from 1 to 7. This will store the 7 Words of data sent from the OCS into the simulated registers %R1 to %R7 in the PC.
5. Create a tag with the address of “C1:1:R1”. Use the OPC Client software to view the data. The OCS is sending the number of seconds from its real-time clock. C1:1:R2 to C1:1:R7 contain the rest of the real-time clock values.
6. Coordinate with another lab group to send and receive data by adding a consumed and produced EGD exchange.

References


WILLIAM LIN
William Lin is currently a faculty member in the Purdue School of Engineering & Technology at IUPUI. Prior to coming to IUPUI, Bill has served as a faculty member at The Pennsylvania State University, Wayne State University, and DeVry University. Before joining IUPUI, he was a Full Professor in EET at DeVry College of Technology in North Brunswick, New Jersey. He has taught and developed various courses in technical areas. Dr. Lin received his Ph.D. in Electrical Engineering from The Pennsylvania State University, a M.S. degree in Physics from the University of Southern Mississippi, and a B.Ed. in Science Education from Taiwan.

MARVIN NEEDLER
Dr. Marvin Needler is a professor of Electrical and Computer Engineering Technology, IUPUI, and has taught engineering and engineering technology for many years.

RICHARD E. PFILE
Richard E. Pfile is a professor of Electrical and Computer Engineering Technology at IUPUI. He received his B.S. from the University of Louisville and his M. Eng. from the University of Michigan. He has won the school Outstanding Teaching Award and has received several Teaching Excellence awards from the School of Engineering and Technology at IUPUI. He teaches courses in microprocessor systems, computer networks and digital signal processing and develops embedded and PC-based software for industry. He has twenty years of teaching experience and eight years of industrial experience, including three years as a systems engineer.

KEN JANNOTTA, JR.
Ken Jannotta has been with Horner APG for ten years and currently manages the PC software development. He is also heavily involved in the firmware and hardware development of their industrial control and networking products. Ken has taught part-time at IUPUI since 2001 and has assisted with curriculum development. He received his degree in Computer and Electrical Engineering from Purdue University.