2006-551: A NOVEL APPROACH TO SIMULATING FACTORY CONTROL SYSTEM PROBLEMS THROUGH A PC AND FOUR MICROCONTROLLERS

Daniel Dangelo, Intel Corp.

Daniel Dangelo is a Test R & D Engineer at Intel Corporation in Chandler Arizona. He manages the CPU Low Cost Burn In Equipment New Product Extensions Development Working Group and provides electrical design and validation support. Dan has worked at four other technology companies during his greater than nineteen years in the industry. Dan has four patents pending and published five papers.

Dan received his Master of Science in Technology and Bachelor of Science in Electrical Engineering from Arizona State University and Associate of Electronics Technology from A.T.E.S. Technical Institute.

Rajeswari Sundararajan, Arizona State University

Raji Sundararajan is an Associate Professor in the Dept. of Electronics and Computer Engineering Technology at Arizona State University, Mesa, AZ. Her teaching interests include instrumentation, power electronics and control systems. Her clinical and research interests are, electrical pulse-mediated drug/gene delivery for cancer and various other diseases, characterization of biological tissues, the long term aging and degradation study of high voltage polymer insulators. Dr. Sundararajan received her PhD in Electrical Engineering (Power/High Voltage) from the Arizona State University, Tempe, AZ. Before that Dr. Sundararajan served as an engineer/scientist at the Dept. of Space, Bangalore, India. She is a Member of ASEE and a Senior Member of IEEE.

Narciso Macia, Arizona State University

Narciso (Ciso) Macia is an associate professor in the Dept. of Electronics & Computer Engineering Technology at Arizona State University, Mesa, AZ. He has served as the Associate Chair in charge of electronics-related academic programs. He received a BS and a MS in Mechanical Engineering from the University of Texas at Arlington, where he specialized in dynamic system, automatic control and fluidics. He also received a PhD in Electrical Engineering from ASU. His dissertation dealt with modeling and identification of the respiratory system. He worked for Honeywell (then AiResearch) from 1975 to 1981, in the fluidics group. In 1981 he left AiResearch and co-founded a small company to developed a medical fluidic device that provided oxygen in an intermittent mode to ambulatory patients. He is inventor/co-inventor of several fluid-control related ideas and holds 3 patents.

He has been involved in advising Junior High and High School students, getting them excited about engineering and technology. He participated in an interdisciplinary, project whose goal was to design and build a cart that would autonomously paint the stripes in a soccer field. Electrical, Mechanical Engineering, Computer Science, and Electronic Engineering and Computer Technology participated in this project.

He is active in fluidics, respiratory mechanics, water filtration and recharge, embedded control, entrepreneurship mentoring, and innovative methods for engineering education. He has published 22 papers and has written two books: the first one on modeling and controlling of dynamic systems, and the second one, an accompanying lab manual. He is a Registered Professional Engineer (Mechanical) in the State of Arizona.

A Novel Approach to Simulating Factory Control System Problems through a PC and Four Microcontrollers

Abstract

Real-World Factory equipment often incorporates various types of communication, interface and control features to provide reliable equipment performance and meet demanding production requirements. This paper discusses a novel simulation project developed to address such common problems as they may be encountered by factory equipment control systems. The reader will gain insight about typical problems and solutions.

The module was written in Visual Basic (VB); provides a PC-Based, control and monitoring, Graphical User Interface (GUI); and interfaces to four industrial microcontrollers. Each of these controllers is from a separate manufacturer and programmed with a different software language, including BASIC-52, assembly, ladder and C. RS-232 protocol serially communicates with the VB program through four serial ports that are provided by a PC-Based RocketPort system. Bi-directional serial communication with both binary and ASCII data was developed to study, document and compare methodologies. The GUI was designed to monitor and control factory simulation activities, inputs and outputs (I/O) and alarm functions via microcontroller serial communications.

The factory simulation layout is comprised of a variety of components to focus on diverse design and assembly challenges. These include a controllable motor driven propeller with rotation tracking capability, a closed-loop self-balancing pendulum, a hand-held pendant, temperature and light sensors with controllable sources, a light tree, motorized cam-controlled switches and a mechanical counter. The GUI enabled excellent serial communication with the four microcontrollers to provide factory simulation control and monitoring. The Microsoft VB MSComm drivers and the RocketPort system provided simultaneous and robust bi-directional serial communications. Both binary and ASCII data communications proved to be effective methods by providing fast and accurate control, alarming and monitoring.

Introduction

Production factories are continuously challenged with providing a large variety of cost competitive products with short manufacturing durations. Fast, flexible and reliable electrical and electronic communication, interface and control systems are crucial for factory processes to run efficiently, provide manufacturing flexibility and deliver a wide range of products. Generally a manufacturing process is an equipment collection developed by various manufacturers and operated with various controller types. A modern flexible manufacturing area requires equipment integration into one seamless system. This is challenging due to combinations of computers, microcontrollers, Programmable Logic Controllers (PLC's) or other controller types. The controllers are programmed and wired to control or monitor sensors, actuators, switches, indicators, alarms and similar devices to provide required system functionality[2].

This paper examines a project devoted to electrical communication, interface and control concepts and issues. It was specifically developed to provide a real-world factory simulation model utilizing four industrial microcontrollers and five software languages. Each represents a factory control system interconnected via the RS-232 serial protocol. The model leveraged simultaneous bi-directional serial communication with a PC-based Graphical User Interface (GUI). The GUI was programmed with Visual Basic (VB) and provided the factory simulation control, alarming and monitoring focal point. Several factory equipment control system problems were encountered and solutions were developed. For in-depth learning, it was decided to use microcontrollers from separate manufacturers and programmed with a different software languages, including BASIC-52, assembly, ladder and C[2].

The system interface and control elements were developed and demonstrated by controlling devices based on GUI selection, sensor levels and switch states. These included a controllable motor driven propeller with rotation tracking capability, a hand-held pendant, temperature and light sensors with controllable sources, a light tree, motorized cam-controlled switches and a mechanical counter. It also included a self-balancing pendulum with closed-loop manual and GUI position control and monitoring capabilities. This proved to be the most difficult simulation subassembly to control, so additional details will be provided later. Two printed circuit boards (PCB's) manually control microcontroller inputs with switches and display output status with LEDs. Two breadboards were incorporated to provide development project support and one contained analog signal conditioning circuits.

The reader will gain insight from the comprehensive factory simulation model since it uncovered many real-world problems. Key factory RS-232 communication scenarios combined with interface and control simulation solutions provide good reference material . The central VB GUI control and monitoring example can be implemented into various controls projects in a reasonable timeframe. The challenge was expanded when it was decided to develop simultaneous serial communications without the use of any off-the-shelf communication package. Also, the serial communications development was expanded to include binary and ASCII character bi-directional communication. A variety of software solutions and data decoding methods were incorporated to handle these options.

Project Scope

The project started by determining how real-world controller communication problems could be understood and applied to microcontrollers[2]. First, the RS-232 serial communications protocol was selected due to its popularity. One main objective was to create a factory simulation with multiple microcontroller types and software languages. The first microcontroller was the Motorola 68HC11 (MC02) programmed in assembly language. Next, the Z-World microcontroller (MC04) based, on the Zilog Z-180 processor was purchased due to the fact that it is programmed with the C language. It was decided that the final two microcontrollers would be the EMAC (MC01) and GE Fanuc Micro-PLC (MC03) microcontrollers. MC01 was programmed with BASIC-52. The MC03 microcontroller is was an appropriate controller since it was programmed in the ladder language. Finally, the fifth software language chosen was Microsoft's Visual Basic (VB), because it is a popular and widely used language that can provide powerful GUI features. The communication between the microcontrollers and PC was handled with the Comtrol RocketPort system. This provided up to thirty-two PC COM ports, which allowed each microcontroller to be connected to a separate COM port. Figure 1 outlines the project signal flow concept, which became the basic system design principle.

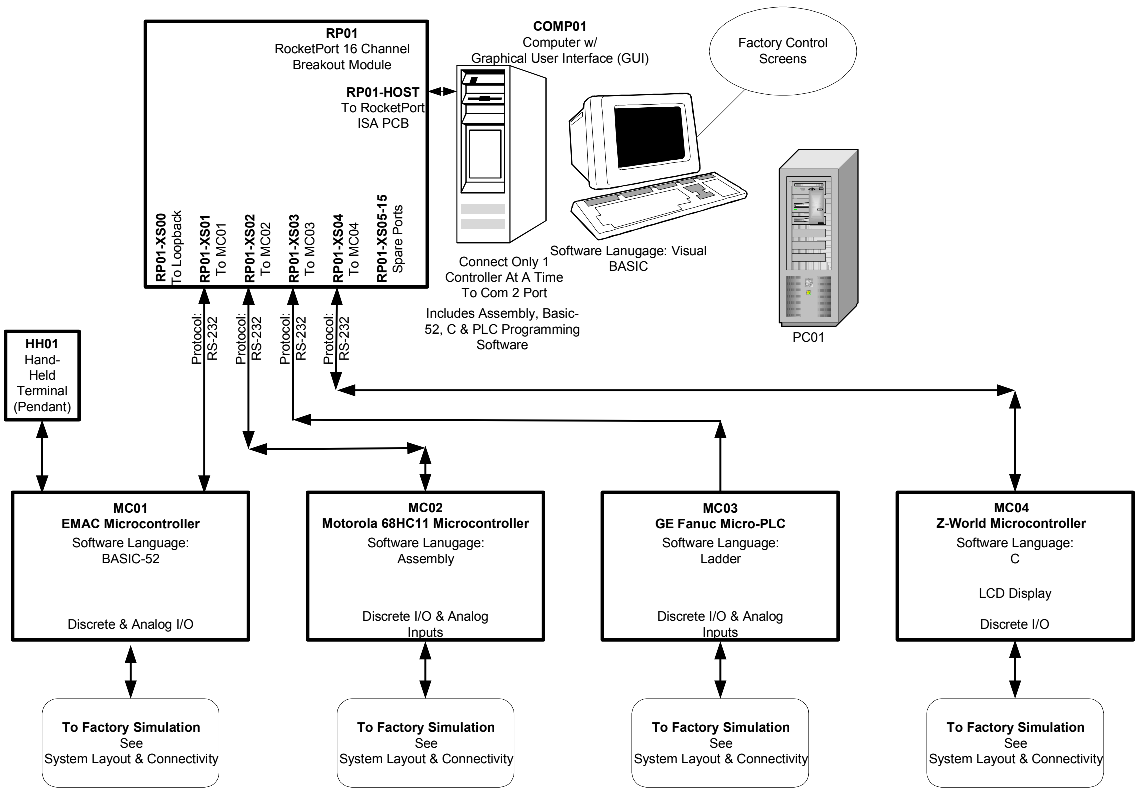

Figure 1. Project Design Concept

Once the main controller system and communication network was established, the next issue was to determine what should be interfaced and controlled. Taking a real-world approach, several components were selected, ranging from industrial components to the motorized cam from a discarded dishwasher. Detailed thought was given to provide actual factory simulation scenarios. For that reason, MC03[3] was chosen to control most simulation devices due to its industrial design features, isolated relay outputs and sinking or sourcing inputs.

The project was then developed on one portable demonstration platform with a removable factory simulation panel shown in Figure 2. The goal was to provide a clean demonstration board and show concepts could be followed during factory equipment development. Cable and wire management was achieved by containing cables and wires in wire trays and sub-surface panel routing. Wire ties were used to bundle loose cables and wires. Spare wires were folded back and heat-shrink was applied to allow for future expansion.

The panel's power supply provided –12, 5, 12 and 24 Volts Direct Current (VDC) distributed through power terminal blocks. It was very important to keep the entire panel's signal commons (0 VDC) connected together. This ensures all components reference the identical 0 VDC to avoid any false differential voltage levels.

A custom printed circuit board (PCB) was developed to interface with the 68HC11 EVBU board Inputs and Outputs (I/O) through factory simulation switching and indication[6]. It was named PCB01 for documentation and reference purposes to improve I/O debugging. Other factory design options were applied such as DIN Rail mounted screw and spring-loaded terminal blocks and a relay to provide quick-mount space saving features to streamline wiring interfaces. One terminal block incorporated a knife-style switch and it allowed the user to manually control propeller power.

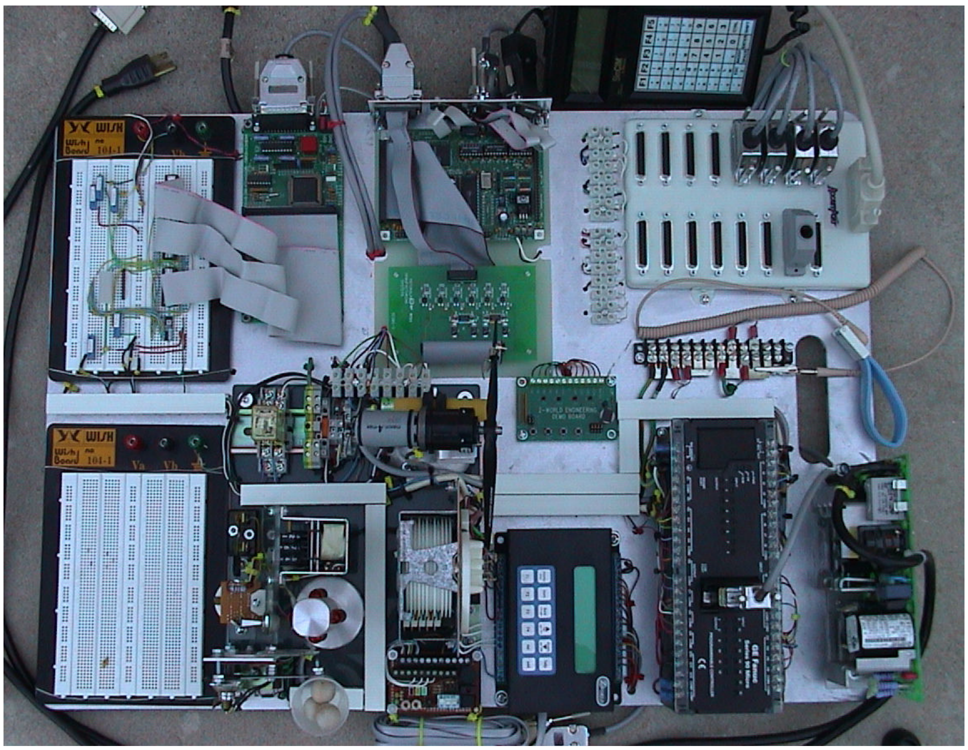

Figure 2. Project Top-View Photograph

Development Key Learning's and Experimental Results

The Comtrol RocketPort[1] system was selected to provide multiple COM Port serial communications to and from the VB program based GUI. Binary and ASCII string data was effectively transmitted and received between the four microcontrollers and VB through the RocketPort[1] system. The initial testing was performed with a loop-back connector to ensure bi-directional COM port operation. The provided software allowed system testing between the software driver, the ISA PCB controller, interface cable, breakout module and loop-back connector. Once that was verified, the next experiments were between two PC's with the RocketPort system connected to one PC and HyperTerminal running on the second PC. The next considerable challenge was to resolve how VB would interface with the RocketPort system. Through research, MSComm was found to be the serial interface solution since Comtrol did not provide VB specific drivers. The RocketPort interface turned out to be easy once MSComm serial communications had been developed. Note MSComm only supports COM ports one to sixteen, which makes RocketPort COM ports seventeen to twenty invalid for VB.

MC03 was found not to have the ability to receive serial data late in the development cycle. This limitation was a surprise for the project and GE Fanuc factory technical representatives. This is understandable due to the large quantity of PLC models GE Fanuc sells. A solution was designed by directly wiring MC04's outputs six to fourteen to MC03's inputs eight to sixteen. Then, the transmitted serial data planned for MC03 was sent to MC04 and transferred to MC03 through discrete I/O lines[7].

Serial data monitors and adapters were very useful with data transfer debugging. The serial monitors use dual-color LED's to reflect data flow, and some include jumper and switch capability to modify signal connections. These monitors and adapters reduced development time and on occasion added problems. Experimentation discovered an in-line monitor caused communication conflicts between MC02 and the PC.

Light tree indicator states were programmed to align with standard industrial machine operations. Indicator definitions can change for different companies and tools, but similarities do remain. See Table 1 for a listing of all light tree states.

Table 1. Programmed Light Tree Indicator States

Color	Type	Description
Green	Solid	System Normal – Ready for normal operation.
Red	Blinking	Unacknowledged System Alarm – Major factory simulation problem occurred. Blinking continues until user acknowledges alarm.
Red	Solid	Acknowledged System Alarm – Acknowledged major factory simulation problem exists. Condition continues until problem is not detected.
Yellow	Blinking	Unacknowledged System Warning – Minor factory simulation problem. Blinking continues until user acknowledges warning.
Yellow	Solid	Acknowledged System Warning – Acknowledged minor factory simulation problem exists. Condition continues until problem is not detected.

Realistic simulations included alarm and warning systems. The warning system monitors hot temperatures and high light intensity events. The VB GUI displays and flashes the warning message and the description of the problem. At that time, the light tree flashes the yellow indicators and PCB01 mounted beeper is activated. GUI and light tree flashing changes to solid indication when warning is user acknowledged, and the beeper is deactivated (silenced) when the warning is acknowledged. The indication disappears only when the warning is acknowledged and the problem is corrected. Factory alarms are activated in cases such as excessive temperature or insufficient light intensity. The alarm functionality is identical to that of the warnings except for the message text content and the light tree indicators are red. The user is also warned to power on the propeller at low speed when the temperature warning appears, and a GUI message directing the user to power the propeller on at high speed is displayed when the temperature is too hot. Additionally, the blue light tree indicators are activated when either the high light intensity or hot temperature warnings are active. The blue indicators signify that user support is required to maintain the system. Required maintenance could include replacing the light bulb or disconnecting the resistive heater.

Several interface problems were encountered, including signal matching and mismatches. Some solutions included moving the signal to another device or changing the signal voltage levels. One unique situation occurred when the double-pole double-throw relay DPDT was used to control the propeller's low and high speed. This is a simple concept of connecting one set of contacts to +12 VDC and signal common and the other to +24 VDC and signal common. The motor power contacts were connected to the relay common contacts. A problem occurred only when the motor speed was changed from high to low. The propeller's speed reduction caused abnormal system problems, motor speed fluctuations and indicator flickering. The problem was quickly isolated to the propeller's motor back feeding the low speed +12 VDC line with the just switched +24 VDC voltage level. Placing an isolation diode in the motor power line easily solved the problem.

The Assembly language based MC02[6] used closed-loop control architecture to operate the self-balancing pendulum (PEND01) with two analog inputs and eight discrete outputs. The output bit status was determined by the difference between the required position and the actual position. Due to being sinking logic, a value of one activates the corresponding discrete output to zero volts. A zero output status bit caused the sinking output level to float at approximately +5 VDC. The GUI sent one force data bit to select between the bread-board mounted potentiometer or VB GUI position control. This bit is controlled by a user-controlled check box. A user controlled slide bar controls the VB GUI pendulum position. The eight discrete output bits are connected to an external 8-bit Digital-to-Analog Converter (DAC) and its analog output signal controls the pendulum mounted power transistor. The power transistor controlled the solenoid current input based on the DAC output level. Two photo-resistors were wired in series to provide the pendulum's position. The photo-resistor was very sensitive to light intensity changes with 1 k Ohms when not blocked and 150 k Ohms when blocked by the pendulum arm. A +5 VDC zener diode was added across MC02's analog input for protection. Fine tuning was enabled with a potentiometer.

The closed-loop control operated as follows. If the error was negative, all DAC outputs were off and the power transistor and solenoid power off. Gravity pulled the pendulum in the opposite direction to balance the system. When the payload increased, the pendulum rotated clockwise, light intensity increased and both Rref resistance and Vfeedback decreased. The control loop corrected this action by calculating the error and applying it to eight MC02 discrete outputs. The outputs drove an eight-bit DAC and it controlled the pendulum's solenoid to rotate the pendulum counter-clockwise. If the error approaches zero then the solenoid current also approaches zero. Figure 3 shows the pendulum electrical connections in schematic format.

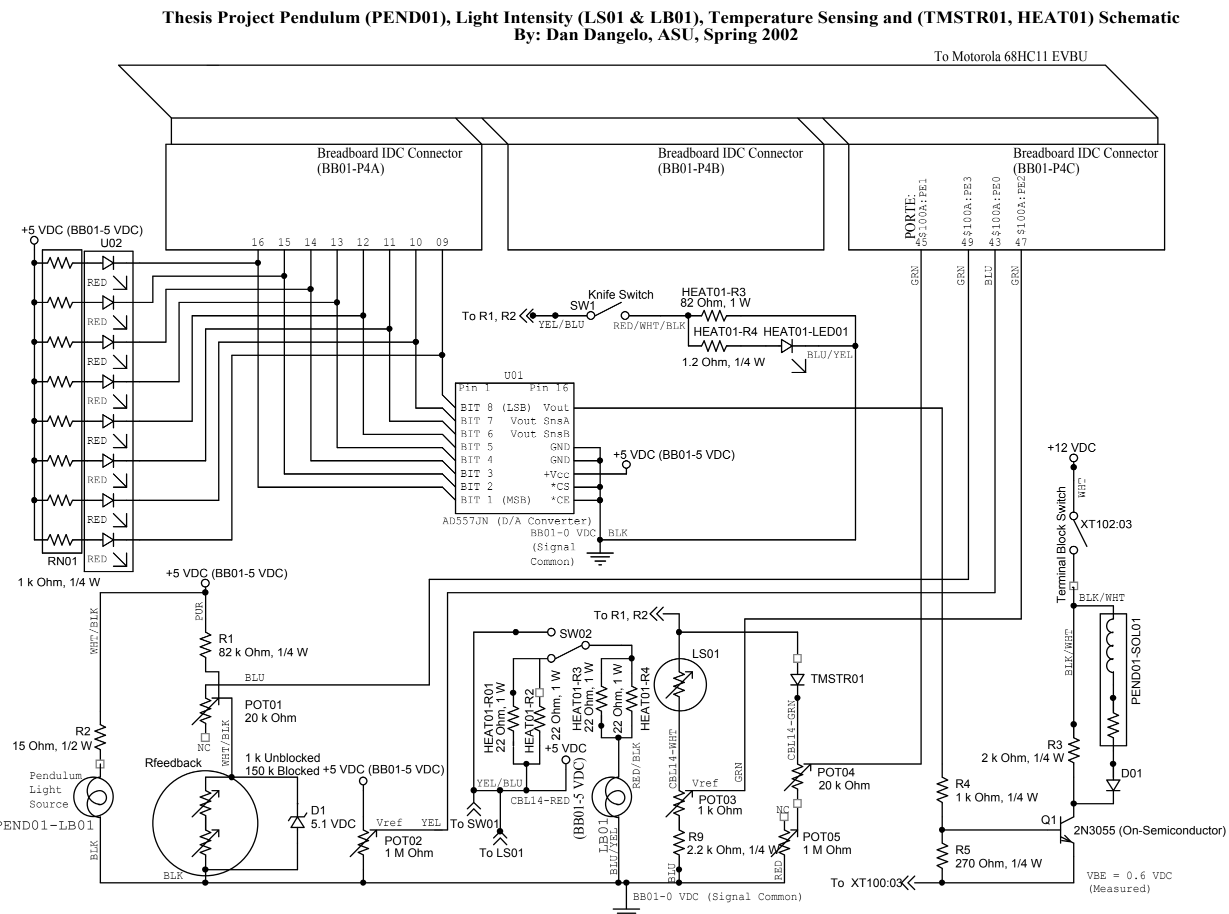

Fig. 3. Self-Balancing Pendulum, Light Sensor and Temperature Sensor Schematic

Electrical Interface Challenges

I/O electrical interfacing can be challenging. For example, sinking refers to zero VDC activating an input or an output's level when activated. Sourcing refers to the input's specified voltage being reached and it is activated. The stated standard rules apply for most computer-based systems. Some companies reverse the terms sinking and sourcing because the source reference is reversed. This is frequently seen in automobile automation designs. At its most basic elemental level, computer based designs reference the component as sinking a signal to zero VDC and not the input sinking the component to zero VDC. Thus, one must use caution when designing a new control system. A sourcing output drives the specified voltage level when the output is activated. Open collector outputs allow a range of activation voltages and are electrically wired to provide required values.

Fortunately, many controllers like MC02 provide configurable sinking or sourcing +5 VDC inputs. But, I/O voltage level differences may require voltage level shifters, relays or optical isolators. During this project, various interface flexibility and problems were encountered. The +5 and +24 VDC components were wired to compatible I/O and MC03 relay output contact isolation. The light tree used LEDs for indication and they required current limiting resistors to allow MC03 output relays to control them with positive +24 VDC. The capacitive proximity sensor (S01) can operate on within a positive +10 VDC to +24 VDC input voltage range, but sinks (drives signal to zero VDC) when an object is sensed. Thus, it could be wired to MC03's +24 VDC input circuit, but not the other microcontroller's +5 VDC input circuits. The inductive proximity sensor (S02) also operates in the +12 to +24 VDC input voltage range.

The temperature thermistor sensor and the light photo-resistor changed resistance values when temperature or light intensity changed. They were wired to voltage divider circuits to interface with the zero to positive +5 VDC MC02 analog inputs. Note the following temperature sensing design rule: metal resistance increases with temperature and semiconductor material (thermistor) resistance decreases. Also, a thermistor is very accurate in small temperature range.

Static electricity dangerous to electronic devices and destroys components when discharges occur in the wrong areas. Therefore, a grounded wrist strap was provided with this project to be worn when working with the components. Grounded cables discussed in the next paragraph help protect systems, but were not there main function.

Electrical noise immunity was improved by following proper earth grounding and cable shielding methods. The PE01 terminal strip was provided exclusively to connect system earth grounds and shielded cable drain wires. PE01's source is the AC power line's (wall outlet) earth ground. This is an important practice to follow when sensitive control signals can be affected by electromagnetic interference (EMI). The earth-grounded shielded-cables intercept most of the EMI noise and drain it to the system earth ground. Grounding methods are debatable, a developing science and treated like black magic. Grounding at the signal source end only was followed for this project. Some engineers will design grounding cables at both cable ends for signals less than 500 MHZ. Care must be taken to route signal cables away from power lines and at a right angle. The angle reduces the EMI to minimal levels and can be understood by following the "Right Hand Rule".

Cables and wire colors were identified and tracked with numbers and color-coding. Yellow identification labels were attached to cables for tracking. System Connectivity and I/O Map documents were developed to track cable and wire details. Industrial standards exist which regulate wire colors; these standards were followed in this project. The standards bring conflicting problems when interfacing components built following European verses United States standards. Simple labels can cause considerable electrical problems such as MC04's "GND" referring to 0 VDC rather than earth ground. Some devices used in this project were wired by following the IEC (CE) standards and others follow the UL standards. Schematic symbols are also debatable. For example, "V" versus "D" can be used for a diode. The following Tables 2 through 4 include informational only examples.

Table 2. Symbol and Wiring Specifications

Abbreviation	Organization Name
ANSI	American National Standards Institute
CE	European Conformity or Conformity Europe
CSA	Canadian Standards Associations
DOD	Department of Defense
EU	European Union Create CE directives
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISA	Instrument Society of America
MIL	Military Standards
NEC	National Electric Code
NFPA	National Fire and Protection Agency
UL	Underwriters Laboratory Inc.

Table 3. Wire Color Examples Referencing IEC and UL Specifications

Abbreviation	Wire Color
BLK	Low Voltage Power Common
BLU	IEC Low Voltage Power Common
BLK	IEC Low Voltage Signals
WHT	Low Voltage Signals
BLK	NEC AC High Voltage Hot Power Line
BRN	IECAC High Voltage Hot Power Line
WHT	NEC AC High Voltage Neutral Power Line
LT BLU	AC High Voltage Neutral Power Line
GRN	NEC Earth Ground
GRN/YEL	IEC Earth Ground

Table 4. Industry Standard Specification Examples

Specification	Description
Wiring, Aircraft, Installation of	MIL-W-5088
Wiring, Guided Missile, Installation of General Specification for	MIL-W-8160
Engineering Drawing Practices	DOD-STD-100
Printed Wiring for Electronic Equipment	MIL-STD-275
Printed – Wiring and Printed-Circuits Terms and Definitions Examples: Block Diagram – Shows circuit information in a more simplified form than the Single Line Diagram. It represents the circuit functions by the means of single lines and rectangular blocks without using graphical symbols or reference designators. Cable Assembly – A cable of a definite continuous length, having one or more ends processed or terminated in fittings, which provide for connection to other items.	MIL-STD-429
Standard General Requirements for Electronic Equipment	MIL-STD-454
Identification Coding and Application of Hook Up and Lead Wire	MIL-STD-681
Dimensioning and Tolerance	ANSI Y14.5
Electrical and Electronics Diagrams	ANSI Y14.15
Graphic Symbols for Electrical and Electronics Diagrams	ANSI Y32.2 (IEEE 315)
Graphic Symbols for Logic Diagrams	ANSI Y32.14
Reference Designations for Electrical and Electronics Parts and Equip.	ANSI Y32.16

Visual Basic Program Overview

The key objective was to write a VB program to simultaneously provide bi-directional communications through four serial COM ports via the RocketPort connected to four microcontrollers[5]. Additional goals included determining, designing, testing and reliably using several forms of serial data communication techniques to identify microcontroller advantages and limitations. This was demonstrated by using ASCII character and binary data bi-direction communications, different data decoding methods, event driven versus polling communication methods, output signal forcing, handling discrete/analog data and more. By successfully utilizing these four different microcontrollers, this project aligned with real-world applications since each posed different difficulties. Figure 4 shows the GUI screen with one alarm active on the right side.

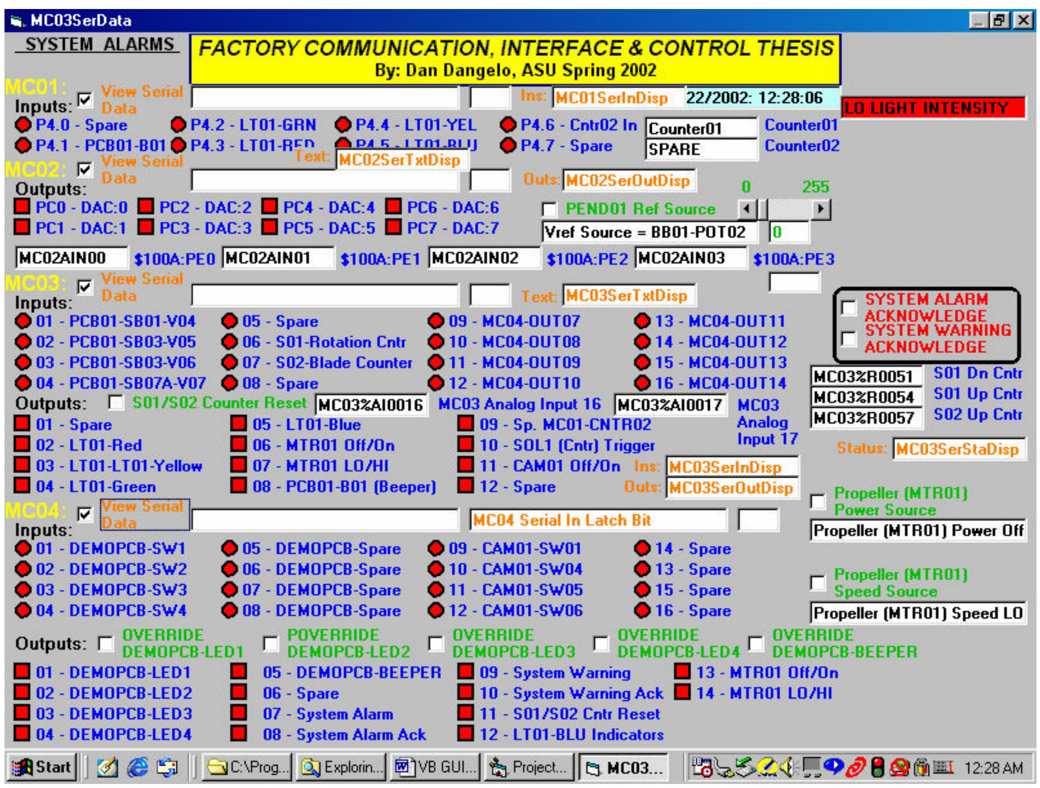

Fig. 4. Screenshot – Project GUI Screen During

Conclusion

This project provided numerous real-world factory communication scenarios and solutions by utilizing the PC-based GUI, four unique industrial microcontrollers and five software languages. The simultaneous bi-directional RS-232 serial communication showed the system's power, flexibility, complexity, problems and solutions. The GUI control, alarming and monitoring enabled complete problem set. The various interface and control additions revealed several ways to enable factory system operation. The unique module and component selections addressed useful features while incorporating closed-loop control, DACs, discrete I/O and indication. The inductive, capacitive, light, mechanical and temperature sensor additions improved the control and monitoring capabilities. The self-balancing pendulum enhanced the hardware and software control system integration capabilities while showing unique operational results.

The project was extremely time consuming and detail oriented. Controller and component reuse would have significantly reduced the effort, but many lessons would have not been learned. This information should be used on future projects to provide a reuse based system with the controllers and components that best fit the application. Reuse will reduce the electrical interface challenges with proper early planning. Many other controllers are available and contain some of the parameters as the ones used in this project. The VB programming language is recommended for GUI development due to its object oriented nature and ease of use.

Bibliography

[1] Comtrol Corporation. RocketPort Multiport Serial Controller [database on-line] (Maple Grove, MN: Comtrol Corporation, N/A, accessed 11 March 1997); available from http://www.comtrol.com/sales/SPECS/rocket.htm or http://www.comtrol.com/products/products.asp; Internet.

[2] Dangelo, Daniel J. 2002. *Factory Communication, Interface and Control* Thesis, Arizona State University, Polytech: May 2002

[3] EMAC, Inc. Single Board Computers Microprocessor Tranining Systems, Custom Engineering & Manufacturing MicroPac 535 [database on-line] (Carbondale, IL: EMAC, Inc., 2000, accessed 21, March 2002); available from http://www.emacinc.com/sbc_microcontrollers/micropac535.htm; Internet.

[4] GE Fanuc Automation North America Inc. *Series 90™ -30/20/Micro Programmable Controllers Reference* Manual, GFK-0467L. rev L. [CD-ROM] (Charlottesville, VA: N/A, 1999).

[5] Microsoft Corporation. Visual Basic 6.0 Documentation. [database on-line] (Redmond, WA: Microsoft Press, 1999, accessed 08 January 2002); available from http://msdn.microsoft.com/library/default.asp?URL=/library/devprods/vs6/vbasic/vbcon98/vbstartpage.htm; Internet.

[6] Motorola Inc. *M68HC11 Universal Evaluation Board User's Manual*. 3rd ed. N/A: Motorola Inc, 1997.

[7] Z-World Inc. *C-Programmable Controller PK2200 Series Condensed Data Sheet Rev B* [database on-line] (Davis, CA: Z-World, Inc., n.d., accessed 06 March 2002); available from (http://www.zworld.com/products/pk2200/docs.html), 1 – 3.