AC 2007-2144: CAN-BASED FIELDBUS EXPERIMENTS

Sri Kolla, Bowling Green State University

Sri Kolla is a Professor in the Electronics and Computer Technology Program at the Bowling Green State University, Ohio, since 1993. He worked as a Guest Researcher at the Intelligent Systems Division, National Institute of Standards and Technology, Gaithersburg, MD, 2000-‘01. He was an Assistant Professor at the Pennsylvania State University, 1990-‘93. He got a Ph.D. in Engineering from the University of Toledo, Ohio, 1989. His teaching and research interests are in electrical engineering/technology area with specialization in artificial intelligence, control systems, computer networking and power systems. He is a senior member of IEEE and ISA.

CAN-Based Fieldbus Experiments

Abstract

This paper presents Controller Area Network (CAN) based hardware experiments, and integration of these experiments in instrumentation and control systems courses laboratories at the Bowling Green State University (BGSU). A general purpose PIC microcontroller based CANstart module is first introduced. This board provides communication features for simple digital inputs such as switches and digital outputs such as LED's. It also has analog inputs and relay outputs. All these devices communicate using CAN protocol through CANoe software. The first experiment familiarizes students with the CANstart module. The second experiment describes interfacing a stepper motor to CANstart board. The final experiment describes the conversion of a closed-loop temperature control system exercise, performed using NI-ELVIS and LabVIEW previously, to CAN communication environment. Interfacing of a temperature sensor and an actuator to CANstart board, and development of a CANoe based software to accomplish the closed-loop control is explained. It is hoped that these experiments will give sufficient hands on experience to students on CAN communication protocol and CAN software and hardware tools in instrumentation and control system courses.

I. Introduction

Fieldbus networks are digital, two-way, multi-drop communication links among intelligent control devices, which are currently used in the industry as a replacement for the traditional 4-20 mA standard. While there are different networks, such as AS-I, Devicenet, Ethernet, Foundation Fieldbus and Profibus [1] available in the industry, Controller Area Network (CAN) is more popular in automotive applications. It is important to integrate fieldbus networks topic in engineering and technology courses so that the curriculum is aligned with the current industrial practice [2]. To emphasize this importance, reference [3] presented advantages of fieldbus networks, a generic communication protocol model and the deviations from this model for various fieldbus networks were identified. As an example of a fieldbus, an overview of CAN was given. CANoe, a CAN simulation software, was outlined, and simulation experiments that are based on CANoe were outlined without much emphasis on hardware experiments in [3].

There is significant literature available on fieldbus networks [1,4]. Hulsebos has been maintaining a comprehensive web site since 1999 that lists various fieldbus networks with links to official web sites of each fieldbus organization [5]. Integration of fieldbus topics into undergraduate curriculum is slowly taking places at various institutions. For example, Franz [6] reported the development of a National Center for Digital and Fieldbus Technology (NCDFT) under an NSF grant at Lee College, Texas. McKean [7] described the evolution of a networking curriculum and integration of CAN and control networks topics in that curriculum. Hong and Kim have developed network based intelligent motor control system using a fieldbus [8]. Also, in Reference [9] Luntz et al. and in Reference [10] Bartz et al., described how internet-based experimentation concepts are adopted in manufacturing and control engineering curriculum. The concepts of fieldbus networks such as Devicenet are also introduced in PLC courses at several institutions [11,12]. An in depth understanding of the literature reveals that there is still a greater need to integrate fieldbus topic with hands-on experiments into undergraduate engineering and technology curriculum.

This paper, therefore, presents CAN-based hardware experiments, and integration of these experiments in instrumentation and control systems courses laboratories at BGSU. Following a brief overview of CAN in Section II and CANoe software in Section III from reference [3], a general purpose PIC microcontroller based CANstart module is introduced in Section IV. This board provides communication features for simple digital inputs and outputs, and analog inputs. All these devices communicate using CAN protocol through CANoe software. Section V describes different hardware experiments developed using CANstart module. The first experiment familiarizes students with different features of the CANstart board. The second experiment involved interfacing a stepper motor to CANstart module. The final experiment is a closed loop temperature control system. Interfacing of a temperature sensor and an actuator to CANstart board, and development of CANoe software to accomplish the closed-loop control is explained. Concluding remarks are offered in Section VI.

II. CAN Overview

The Controller Area Network (CAN) standard, popular in automotive applications, defines a simple broadcast serial network that works well for real-time short-range communications [13-15]. Bosch developed the CAN protocol, which has since been standardized internationally as ISO11898 and has been "implemented in silicon" by several semiconductor manufacturers. CAN is the basis of several sensor buses such as Devicenet, CANopen, J1939, and Smart Distributed System.

CAN uses a twisted pair cable to communicate up to 40m at speeds 1Mbit/s without repeaters, and up to 1 km at 20 kbps speed. It can support up to 40 devices. CAN uses CSMA bus arbitration. The CAN protocol, which corresponds to the data link and physical layers in the ISO/OSI reference model [1], meets the real-time requirements of automotive applications. CAN data packets are 8 bytes long and use 11-bit packet identifier. A second version of CAN can support 29 bit identifier.

Each CAN data frame consists of seven different bit fields shown in Figure 1. A data frame begins with the start-of-frame (SOF) bit. It is followed by an eleven-bit identifier and the remote transmission request (RTR) bit. The identifier and the RTR bit form the arbitration field. The control field consists of six bits and indicates how many bytes of data follow in the data field. The data field can be zero to eight bytes. The data field is followed by the cyclic redundancy checksum (CRC) field, which enables the receiver to check if the received bit sequence was corrupted. The two-bit acknowledgment (ACK) field is used by the transmitter to receive an acknowledgment of a valid frame from any receiver. The end of a message frame is signaled through a seven-bit end-of-frame (EOF). Further details of CAN such as arbitration and error handing can be found in many references [13].

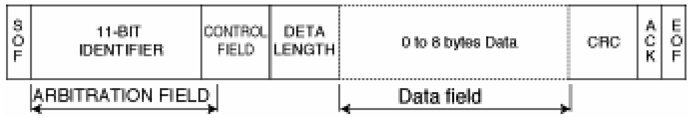

Figure 1. CAN Data Frame.

III. CANoe Simulation Software

CANoe, developed by Vector CANtech, is a robust CAN tool that is capable of simulating a CAN system [16]. CANoe supports the entire development process for networked systems from planning to implementation. CANoe offers special functions for all phases of the development process of distributed systems and its Electronic Control Units (ECUs), e.g. model creation, simulation, functional testing, diagnostics, and analysis.

CANoe supports the test of ECUs and networks via special functions of the Test Feature Set (TFS). With these functions, tests can be created to verify single development steps, check prototypes or execute regression and conformity tests. Additionally, the check and stimulus functions, included in the Test Service Library (TSL), simplify the setup and execution of its own test scenarios.

The CANoe functionality can be expanded as desired. Blocks may be inserted at any point in the data flow diagram, and its function can be programmed. The application-oriented, C-like language CAPL (Communication Access Programming Language) is used for this purpose. CANoe includes an interactive development environment, which makes it easy to create, modify, and compile CAPL programs.

Network node models are added to the simulation setup as CAPL programs. These can be created manually or automatically from the database. The Panel Editor and Panel Generator support in creating graphic user control and display panels for the network node models. With system tests, it is often the case that peripheral signals of ECUs must be accessed. This is achieved by reading-in or outputting these signals over a port as environment variables, and these are used in the simulation.

A model for the network system can be created in CANoe using three steps:

1. Create the database with messages, signals and environment variables. CANdb++ editor is used to create these databases.

2. Create the network node periphery, which includes control panels. Panel Editor is used to create these panels.

3. Create the network node model in CAPL program.

Once the system is created, the program is executed and performance characteristics are monitored. The measurement setup windows such as trace window and statistics window allow the observation of these characteristics. Figure 2 shows an automobile simulation, created with CANoe, using the above three step process.

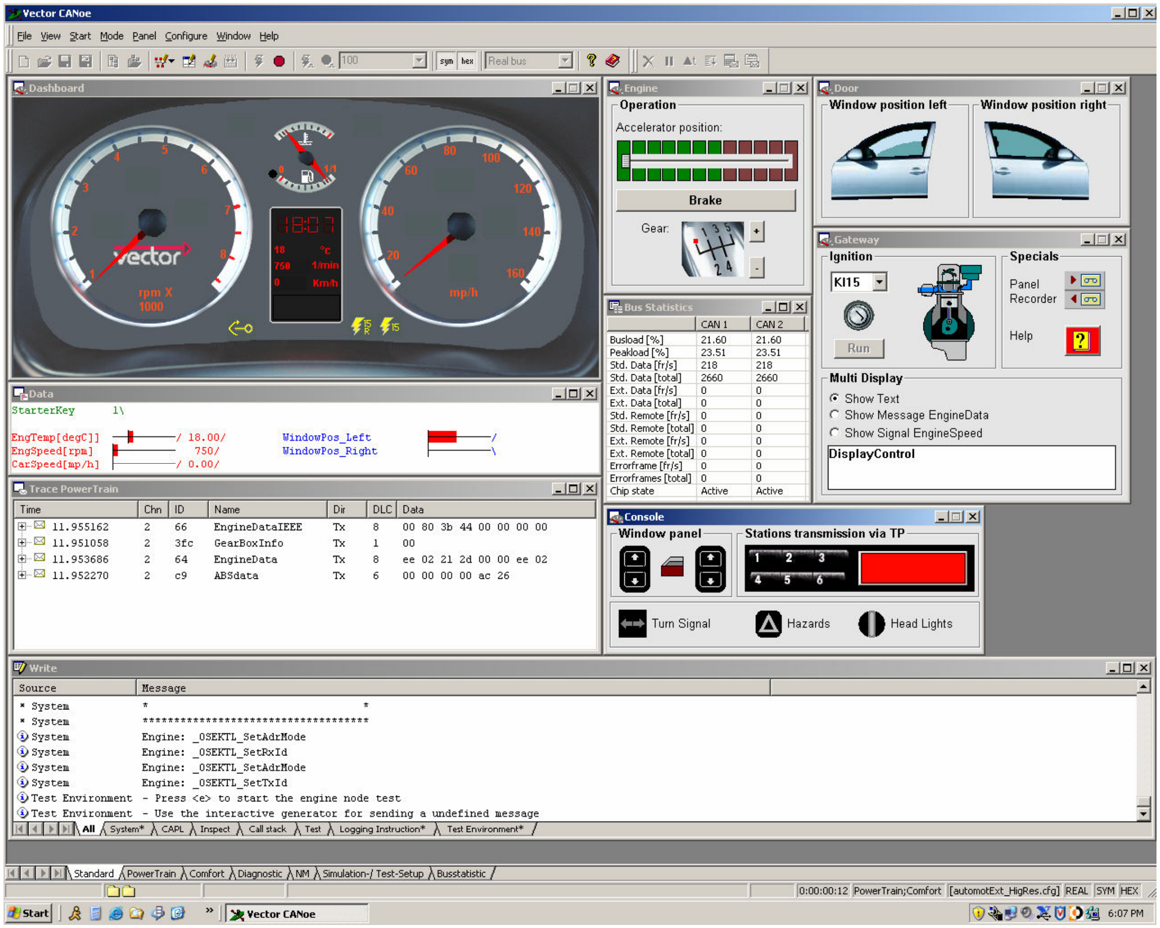

Figure 2. CANoe Simulation of an Automobile Application.

IV. CANstart Module

CANstart, developed by Nuvo Technical Services, is a simple CAN-based electronic module used to enhance the CAN learning experience. The block diagram of CANstart module is shown in Figure 3. A microchip PIC18F458 microcontroller controls all on board functions, interfaces to the user input/output circuits, and handles the CAN communication interface [17]. The PIC microcontroller is pre-programmed and is ready for CAN communication once properly connected to another CAN node such as CANoe. Key features of the CANstart include:

- 4 input pushbutton switches
- 4 user-controlled LED indicators
- 4 digital inputs
- 4 digital outputs
- 4 analog inputs

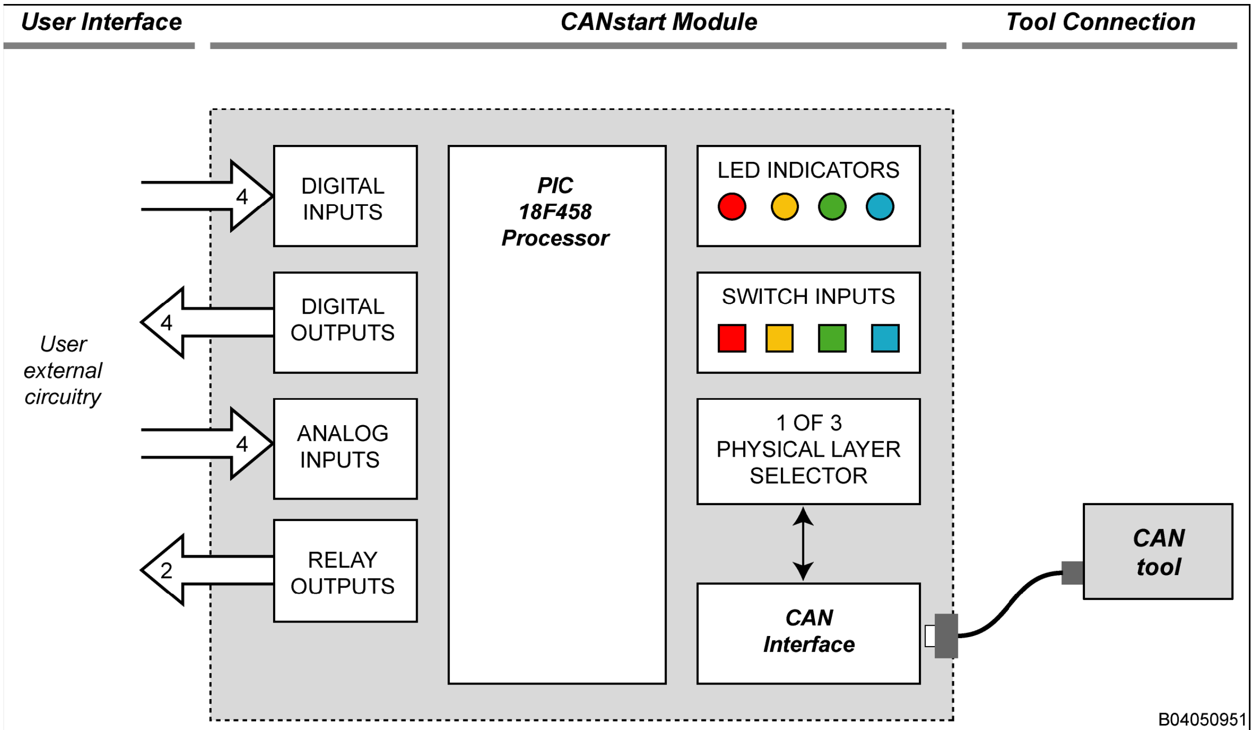

Figure 3. Block Diagram of CANstart Module.

Aside from few on-board DIP switch settings, almost all of the CANstart I/O functions are interconnected to CAN messages. For the on-board CANstart user inputs, the PIC microcontroller reads these values and periodically packages the data information into two distinctive CAN transmit messages. For controlling and activating CANstart's user outputs, a CAN message received by the CANstart module will result in the activation of the commanded output function. To activate CANstart outputs, a CAN tool capable of transmitting a specific command message, such as CANoe, to the CANstart module is required. All information passed to and from the CANstart board is accomplished with three CAN messages shown in Figure 4.

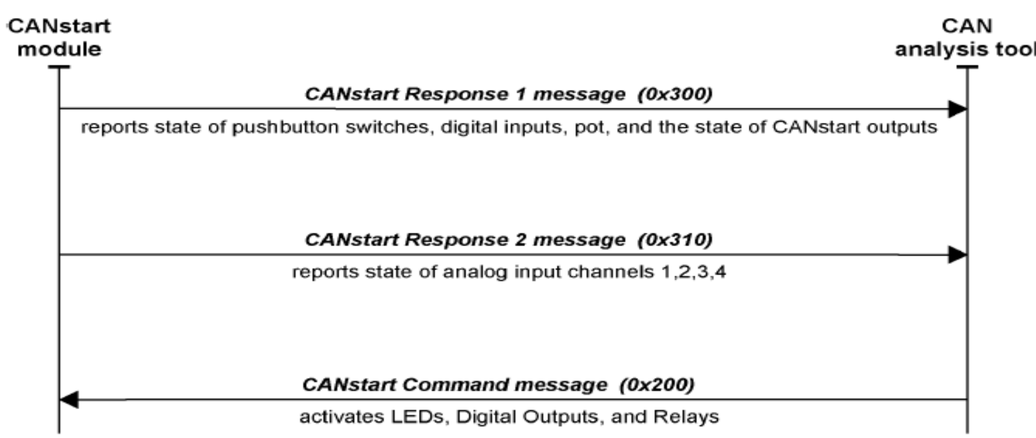

Figure 4. CANstart Communication Format.

Two messages are periodically transmitted by the CANstart board and contain data sampled by various CANstart hardware inputs. A single message is used to send command data to the CANstart board.

V. CAN Based Fieldbus Laboratory Experiments in Instrumentation and Control Courses

Several laboratory experiments are being developed using the CAN hardware and integrated into instrumentation and control courses at BGSU. These experiments use CANstart module. Some of these experiments are described in this section. The first experiment familiarizes students with the CANstart module. The second experiment interfaces a stepper motor to the CANstart module. The third experiment is a closed-loop system for a temperature control process.

V.1 Familiarization with CANstart Module

In this experiment, students get familiarized with the CANstart module and communicate with it using CANoe software. A sample CANoe program allows students to experiment with the various input and output functions. CANstart Response Panel shows the information that is being sent by the CANstart module to the CANoe. CANstart Command Panel shows the information that can be sent by the CANoe to the CANstart module. Figure 5 shows the CANstart board and both these panels running in CANoe environment.

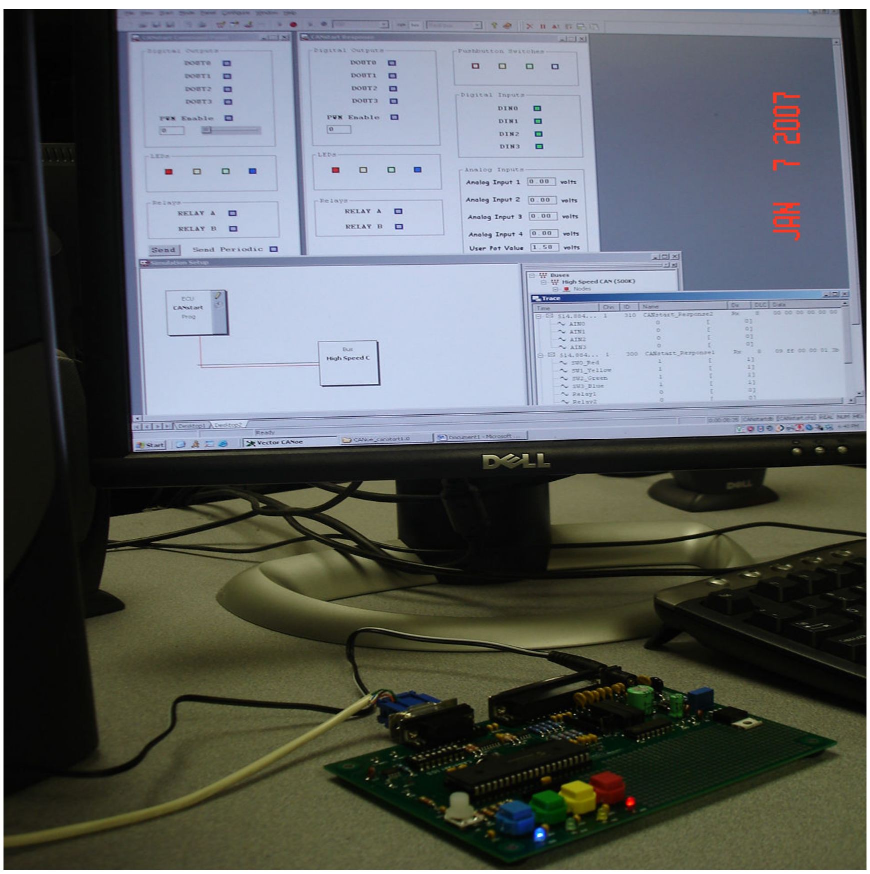

Figure 5. CANstart Module Familiarization with CANoe Panels.

The screen shot of the CAPL program in the simulation setup of CANoe is shown in Figure 6. Students can set desired output states on the Command Panel and send the information to CANstart once by clicking the send button once, or send the information continuously by selecting send periodic option. Students experiment with the four analog inputs by connecting different voltage signals to DB25 connector. Students also experiment with digital outputs and relay outputs by connecting different physical devices.

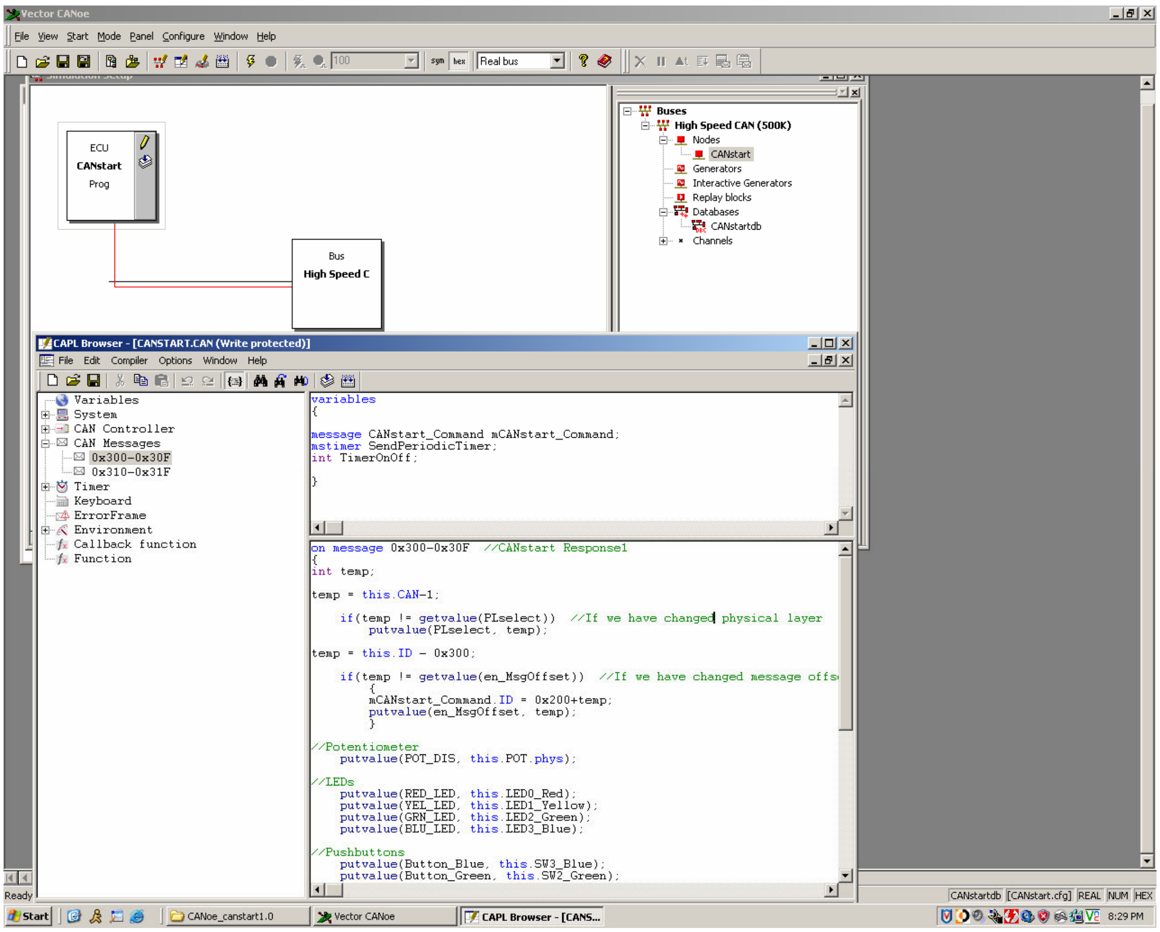

Figure 6. CANoe Simulation Setup for CANstart Module with its CAPL Program.

V.2 Stepper Motor Experiment

In this experiment, students interface a stepper motor to a CANstart module. The stepper motor used in the experiment is a small 100 steps/revolution unipolar stepper motor mounted on a base with a dial pointer and numeric faceplate shown in Figure 7. The motor is connected to the CANstart board using a DB25 connector. The motor control is accomplished through CANoe using the interrupt timer provided by CANoe for controlling the motor step sequence at a regular rate. The program is written so that when the red button on the CANstart board is pressed the pointer moves clockwise, and the pointer moves counter clockwise when yellow button is pressed. Figure 8 shows the measurement setup windows such as trace window and statistics window when the CANoe program is executed to show various performance characteristics. This experiment simulates an automotive speedometer.

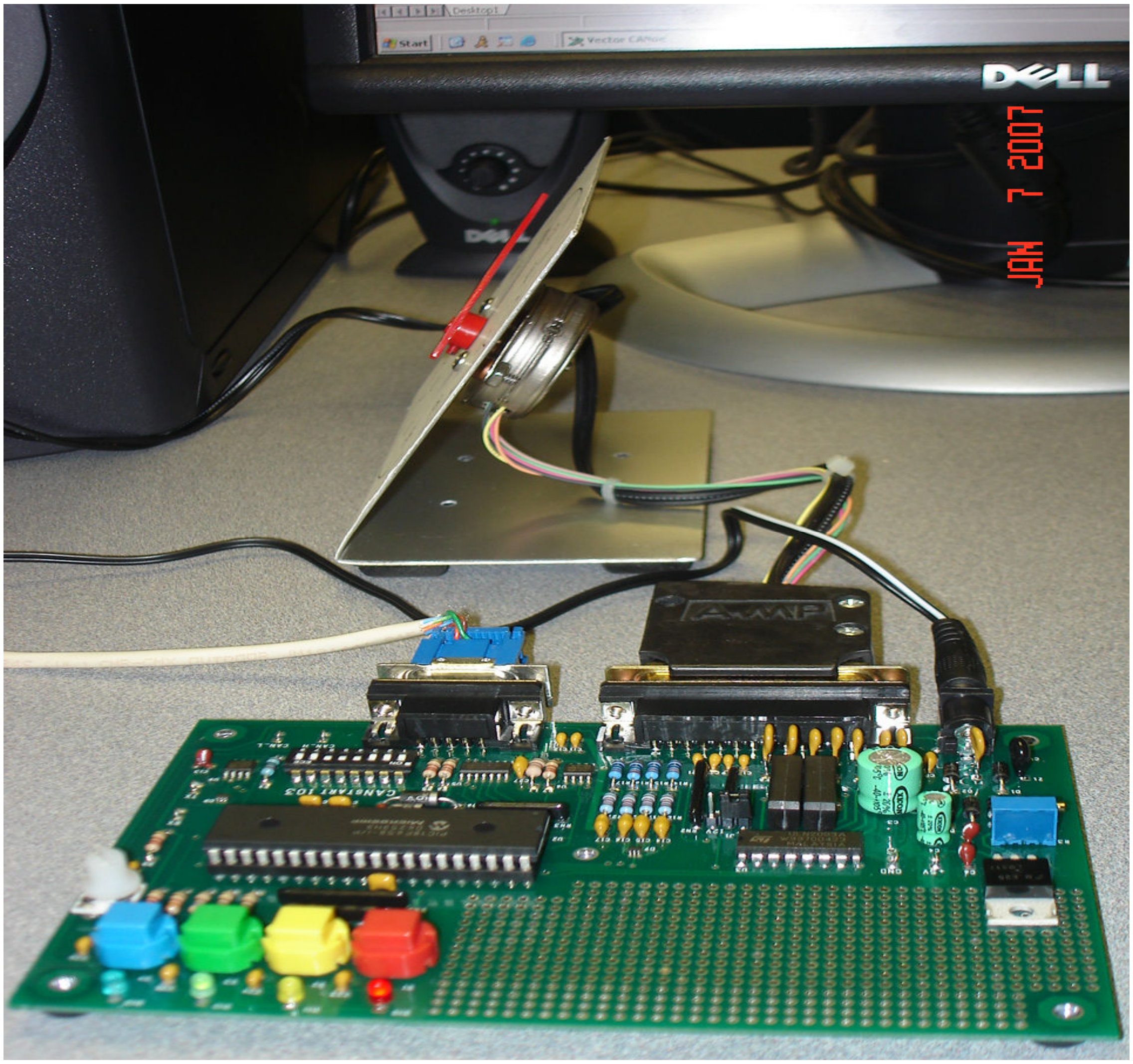

Figure 7. Stepper Motor Interface to CANstart Module.

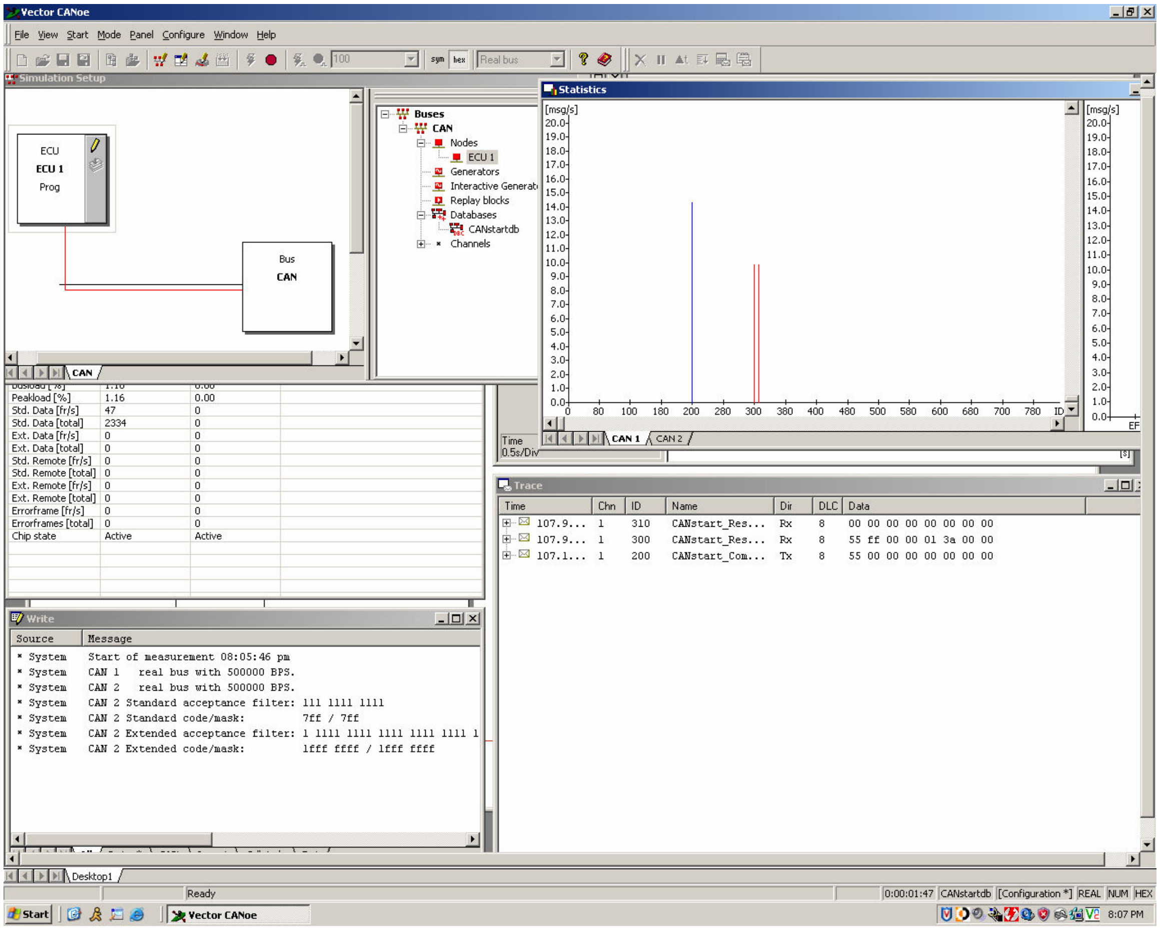

Figure 8. CANoe Communication with Stepper Motor through CANstart Module.

V.3 Temperature Control Experiment

The objective of the temperature control experiment is to maintain the temperature inside a wooden box at some desired set-point value, within neutral zone limits, using a two-state-controller mode [18]. The wooden box is heated with a light bulb. The temperature is measured using LM34 solid-state temperature sensor based circuit, which gives 10mV/°F. A signal conditioner is designed to modify the voltage corresponding to 50 to 150 °F to analog input voltage range of 0 V to 10 V. The output of the signal conditioner circuit is connected to an analog input channel on the CANstart module. When the temperature differs from the set-point value (with ± neutral zone), it results in the high-limit and the low-limit, and a fan is turned ON and OFF accordingly. A digital output on the CANstart module is connected to a solid-state relay (SSR) that controls the operation of the fan. Figure 9 gives a schematic diagram of this experiment.

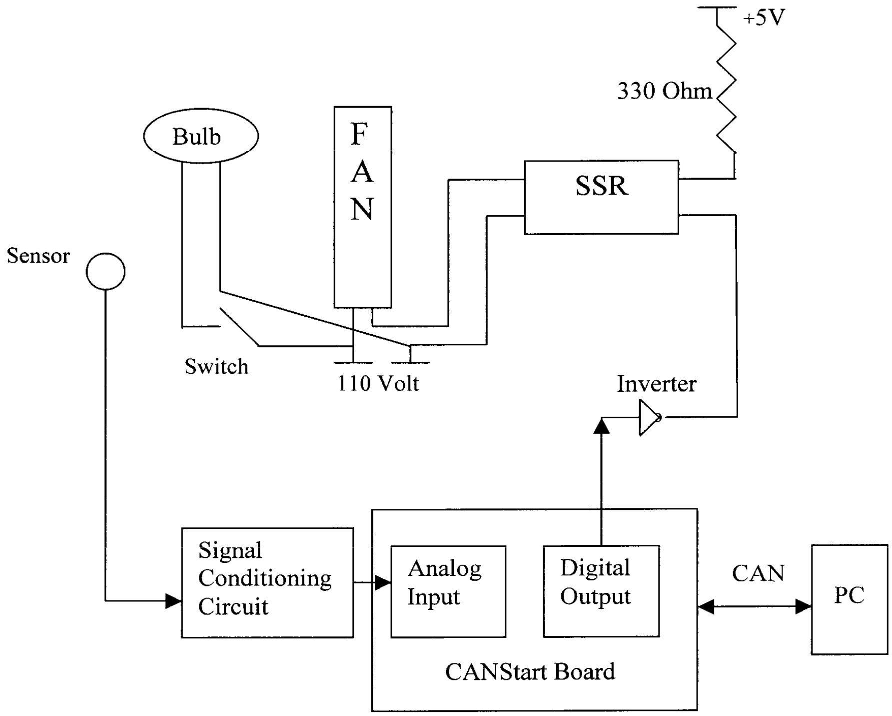

Figure 9. Schematic Diagram of the Temperature Control Experiment.

This closed loop temperature control system exercise was previously performed in the controls course at BGSU using hardware built with NI-ELVIS station and software developed using LabVIEW. In the current experiment, hardware was built for temperature sensor and signal conditioner, and the voltage signal representing the temperature is connected to one of the analog input channels on the CANstart board. A 110 V ac Fan is turned ON/OFF using an SSR controlled by a digital output on the CANstart board. The experiment hardware is shown in Figure 10. A CAPL program developed on CANoe software accomplishes the closed-loop control by appropriately turning the fan ON/OFF to keep the temperature in the box at a desired value. This experiment simulates the air-conditioner operation in an automobile. It will enable students to understand how CAN communication environment is used in closed-loop control applications.

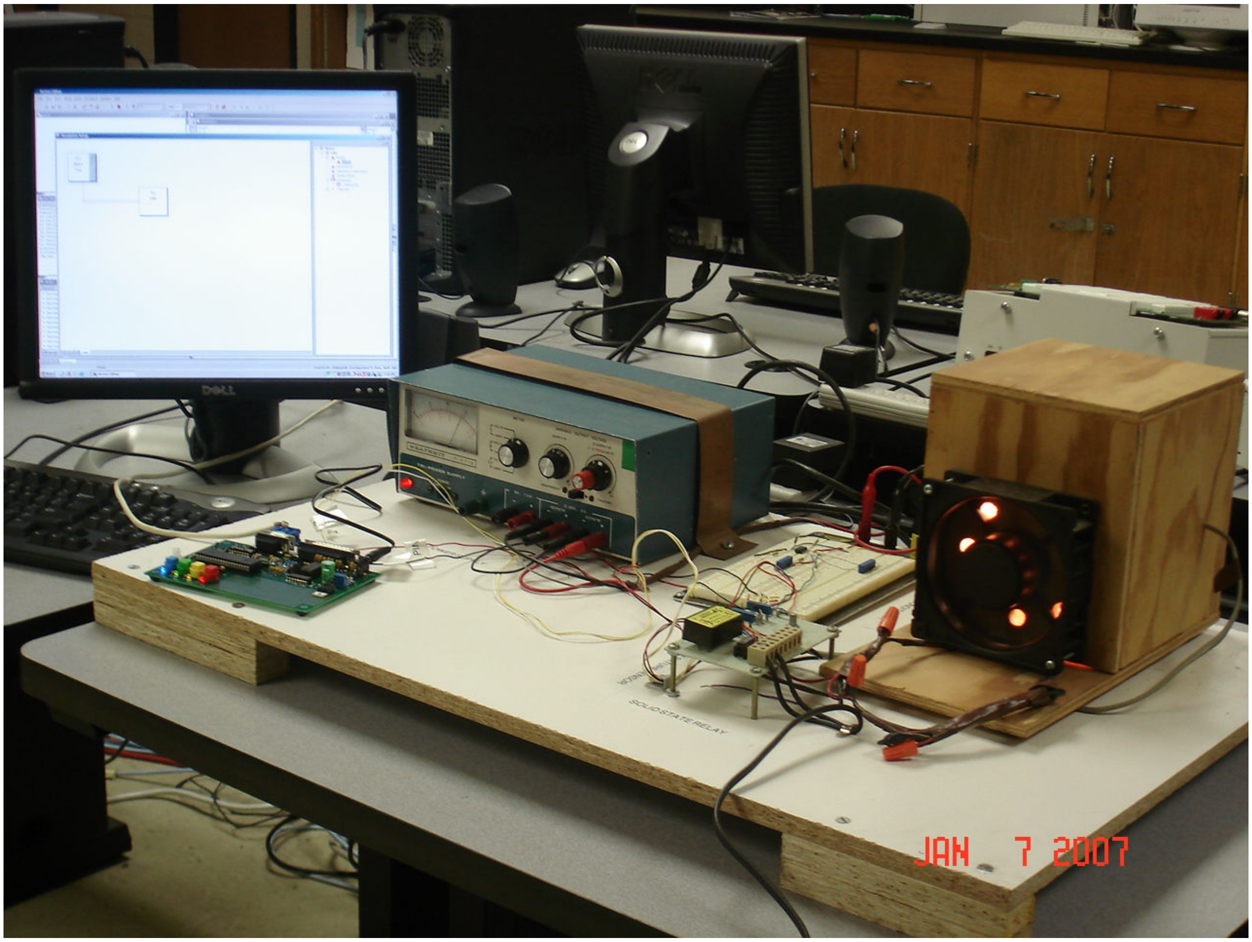

Figure 10. Temperature Control Experiment Hardware.

VI. Conclusions

The purpose of this paper was to present the integration of CAN-based fieldbus experiments into instrumentation and process control course laboratories at BGSU. A general purpose PIC microcontroller based CANstart board was first introduced. Various analog and digital inputs/outputs on this board communicate using CAN protocol through CANoe software. In the first experiment, students get familiarized with different features of CANstart board. The second experiment described interfacing a stepper motor to CANstart board. The last experiment implemented a closed loop temperature control system. Interfacing of a temperature sensor and an actuator to CANstart board, along with CANoe communication was explained. It is hoped that these experiments will give sufficient hands on experience to students on CAN communication protocol and CAN software and hardware tools in instrumentation and control system courses. Several other experiments are being developed for future implementation, such as LVDT-based position measurement exercise and closed-loop control of a dc motor exercise.

Acknowledgements

The author thanks Mr. Bob Matteson of Nuvo Technical Services, Ann Arbor, MI for allowing him to use material related to CANstart module in this paper. The author also thanks Mr. Bruce Emaus of Vector CANtech, Novi, MI for providing support to ECT program at BGSU through CAN Goes to College grant program, and allowing him to use CANoe material in this paper. An Ohio Board of Regents (OBOR) equipment grant provided partial support to this work.

References

[1] Jonas Berge, *Fieldbuses for Process Control: Engineering, Operation, and Maintenance*, ISA - The Instrumentation, Systems and Automation Society Press, Research Triangle Park, NC, 2002.
[2] L. D. Feisel and A, J. Rosa, "The role of the laboratory in undergraduate engineering education," Journal of Engineering Education, January 2005, pp. 121-130.
[3] Sri Kolla and Joseph Mainoo, "Fieldbus Networks Topic in Instrumentation and Control Systems Courses," Proceedings of ASEE Annual Conference, Chicago, IL, 2006
[4] G. Cena, L. Durante and A. Valenzano, "Evolution of standard fieldbus networks," in *Intelligent Systems and Robotics* (G. W. Zobrist and C. Y. Ho, Eds.), Gordon and Beach Science Publisher, 2000.
[5] R. A. Hulsebos, "The fieldbus reference list," http://ourworld.cs.com/_ht_a/rahulsebos/Links.htm, 1999-2005.
[6] Harry Franz, "Fieldbus Instrumentation Technology Development at Houston, Texas Area Universities and Colleges," Proceedings of ASEE Annual Conference, 2003.
[7] Ron McKean, "The evolution of a technical curriculum," Proceedings of ASEE Annual Conference, 2006.
[8] W.-P. Hong and D.H. Kim, "Networked based intelligent motor control systems using IEEE/EIA 709.1 fieldbus," Proceedings of ACIT – Communication Systems Conference, Novosibirsk, Russia, June 2005.
[9] J. Luntz, E. Almeiada, D. Tilbury, J. Moyne and K. Hargrove, "The distributed reconfigurable factory testbed (DRFT): A collaborative cross-university manufacturing system testbed," Proceedings of ASEE Annual Conference, 2006.
[10] R, Bartz, S. Engell, C. Schmid, H. Roth, N. Becker and H. M. Schaedel, "Project-oriented internet-based learning in the field of control engineering," Proceedings of ASEE/SEFI/TUB Annual Colloquium, 2002.
[11] J. Rehg and B. Muller, "Teaching PLCs with the IEC 61131 standard languages," Proceedings of ASEE Annual Conference, 2005.
[12] D. Wang, and H. Peddle, "System approach for design and construction of PLC training laboratory," Proceedings of ASEE Annual Conference, 2001.
[13] K. H. Johansson, et al. (2005). "Vehicle applications of Controller Area Networks." To appear in 2005 in the Handbook of Networked and Embedded Control Systems, D. Hristu-Varsakelis and W. S. Levine, Eds., Retrieved January 10, 2007 from http://www.md.kth.se/RTC/Papers/VehicleApplicationsCan2005.pdf.
[14] H. Kopetz, "A comparison of CAN and TTP." Retrieved on January 10, 2007 from http://www.md.kth.se/RTC/RTCC/Material01/Papers/kopetz-ttpvscan.pdf.
[15] CAN protocol training notes, Vector CANtech Inc., 2004.
[16] CANoe Manual, Version 5.1.1, Vector CANtech Inc., 2005.
[17] CANstart User Manual, Version 1.0, 2006.
[18] Curtis Johnson, *Process Control Instrumentation Technology*, Prentice Hall, 2006.