# Supervisory Control and Data Acquisition Laboratory

Mihai Iacob, Cristina Anita Bejan, and Gheorghe-Daniel Andreescu, *Senior Member, IEEE*

*Abstract* — **This paper presents practical laboratories for teaching purpose in Supervisory Control and Data Acquisition (SCADA) systems. A SCADA system is described in terms of architecture, process interfaces, functionality, and application development facilities. These concepts are implemented on an integrated automation system, particularly for digital control of electric drives with a distributed peripheral, i.e., Totally Integrated Automation with Democase from Siemens. Using this system, a wide range of applications can be designed, implemented and tested. A practical labs set is presented to introduce gradually the main SCADA elements, and finally to develop an application to control an induction motor in interlocked manual/automatic mode, with touch-screen Human Machine Interface (HMI). The system employs industrial busses like PROFIBus and industrial Ethernet. The SCADA system also shows trends, alarms, motor frequency and automatic sequence of motor speed profile.**

*Keywords* — **Automatic control, engineering education, ethernet, hardware configuration, laboratory works, profibus, programmable logic controller (PLC), supervisory control and data acquisition (SCADA), WinCC Flexible.**

## I. Introduction

SCADA systems became popular to arise the efficient monitoring and control of distributed remote equipments. Today SCADA systems include operator-level software applications for viewing, supervising and troubleshooting local machines and process activities. Powerful software technologies are used for controlling and monitoring equipments in easy-to-use web-based applications, e.g., platforms: PCS7 – Siemens, CX-Supervisor – Omron, Genesis 32 – ICONIX [1].

The term *supervisory control* is associated with (i) the process industries, where it manages the activities of a number of integrated operation units to achieve certain economic objectives for a process; and with (ii) the discrete manufacturing automation, where it coordinates the activities of several interacting pieces of equipment in manufacturing cells or systems, such as a machines interconnected group by a material handling system [2].

SCADA encompasses the collecting of information transferred to the central site, carrying out any necessary analysis and control, and then displaying that information on a number of operator screens or displays. The required control actions are then conveyed back to the process [3].

Control and supervision tasks of industrial plants are distributed over wide areas, and are characterized by many sensing and actuation points (of the order of hundreds or even thousands of units) as in petrol chemical plants, paper factories, newspaper rotary printing presses, plants for extraction and bottling of alimentary juices, energy monitoring, etc. They require the use of sophisticated automation schemes that must be able to grant access to production data and field distributed variables at a large distance, and from various levels of factory automation (field, control, supervision, etc.) [4].

*Programmable Logic Controls* (PLCs) are used for system control. As the need to monitor and control more devices in the plant grew, the PLCs are distributed and the systems became more intelligent and smaller in size.

In a *distributed control system* (DCS), the data acquisition and control functions are performed by a number of distributed microprocessor-based units situated near to controlled devices or by instruments from which data are gathered. DCS have evolved into systems providing very sophisticated analog control capability [5].

SCADA, PLC and DCS are three types of control systems. Nowadays, there are systems that incorporate all these concepts in one integrated automation system.

Totally Integrated Automation (TIA) is the foundation to implement industry specific automation solutions that are coordinated with individual requirements, combining increased productivity with a high level of investment security [6]. TIA is the unique basis offered by Siemens for uniform and customized automation in all sectors of the production, process and hybrid industries. TIA offers a uniform automation technology on one single platform for all applications of process automation, starting with input logistics, covering production or primary processes as well as secondary processes, up to output logistics.

This paper introduces theoretical and practical concepts regarding SCADA for education purpose, illustrated and implemented on an integrated automation system, TIA from Siemens. Laboratory works will provide for student: competences, experience in identification, selection and programming of needed equipments to make a fully

One of authors, Ph.D student Mihai Iacob thanks for financial support of his work under strategic grant POSDRU 6/1.5/S/13, 2008 of the Ministry of the Work, Family and Social Protection, Romania, co-financed by the European Social Found.

Mihai Iacob is with “Politehnica” University of Timisoara, Dept. of Automation and Applied Informatics, Timisoara, Romania (e-mail: mikyblue00@yahoo.com).

Cristina Anita Bejan is with “Politehnica” University of Timisoara, Dept. of Automation and Applied Informatics, Timisoara, Romania (e-mail: bejananita@gmail.com).

Gheorghe-Daniel Andreescu, is with “Politehnica” University of Timisoara, Dept. of Automation and Applied Informatics, Timisoara, Romania (e-mail: daniel.andreescu@aut.upt.ro).

operational application. The emphasis is placed on the study of the concepts and languages of control, to guide the student on how to analyze and design such systems.

Terminology, concepts, principles, procedures, computations, and communication protocols used in the control field programming are studied. The programming of the human machine interface (HMI), to present process data to a human operator, is another subject of interest.

During the practical labs, in addition to understanding the theoretical concepts, the students will develop applications to illustrate them. Finally, the application refers to control of an induction motor associated with a frequency inverter, in a manual or automatic mode. The manual control mode is equivalent to predefined frequencies and logic programming that are sent as telegrams to the frequency inverter. In the automatic control mode a specific sequence is run (sequential discrete control).

## II. Practical Labs for Scada Systems Using TIA Democase

Using the equipment from Siemens SCADA – TIA Democase, the students will develop practical labs for a real functional application.

The development platform for applications is STEP 7, which is the standard software package used for configuring and programming SIMATIC programmable logic controllers. The standard package includes a series of valuable tools like: Symbol Editor, NETpro, SIMATIC Manager, Hardware Configuration, Programming Languages (Ladder Logic (LAD), S7Graph) and Hardware Diagnostics.

To create an automation solution with STEP 7, there are a series of basic tasks:

- Plan your controller;
- Design the program structure;
- Create a project structure;
- Configure hardware;
- Configure networks & communication connections;
- Define symbols;
- Create the program;
- Configure operator control and monitoring variables
- Download programs to the programmable controller;
- Test programs;
- Monitor operation, diagnose hardware;
- Document the application.

A project contains all programs and data for the entire automation tasks. It is an object structure, under which other STEP 7 objects will be organized [7]. This project can contain several hardware stations, networks or several programs that are used in one or several CPUs.

Following these steps, through practical labs, the students will learn to use the equipment and program it, and the final result will be an application for controlling an induction motor. Each practical lab has the following structure: *purpose, theoretical topics, workflow, questions and references.*

The gradual laboratory sections are structured as: A) General description of SCADA - Totally Integrated Automation Democase, B) Hardware configuration and programming languages, C) Programming and commissioning for frequency inverter SINAMICS G120, D) Manual control for frequency inverter, E) Automatic control for frequency inverter, F) Integration of manual and automatic control modes for frequency inverter, G) Introduction to WinCC Flexible SCADA platform, H) Programming elements in WinCC Flexible. These lab sections are shortly presented in what follows.

### *A. General Description of SCADA - Totally Integrated Automation Democase*

The aim of this section is to present the SCADA system with its components. SCADA - TIA Democase from Siemens (Fig. 1) is an integrated automation system for digital control of electric drives with a distributed peripheral [8]. It can be used for a wide range of applications with variable speed drives, like: pumps, fans or conveyor belt, etc. This subject covers applications to plant and mechanical engineering, as well as manufacturing technology and process engineering.

The main TIA Democase integrated components are:

- S7-300 PLC that controls ET200S;
- ET200S consists of: communication interface module IM 151-3, Motor starter DS1e-xH, proximity sensor Opto Bero;
- Sinamics G120 frequency inverter;
- induction motor connected via PROFIBUS;
- *Human Machine Interface* (HMI) panel MP 277 connected via industrial Ethernet.

The motor is controlled manually with predefined frequencies, and automatically by sequence.

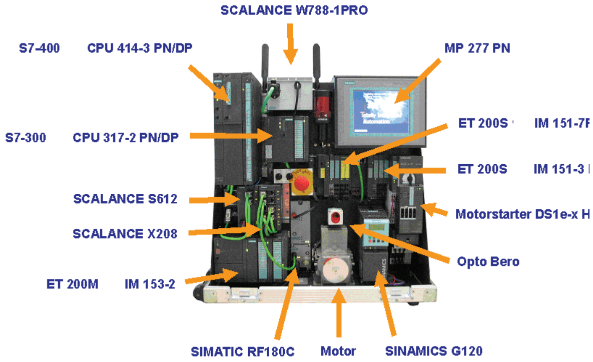


Fig. 1. SCADA – TIA Democase setup.

### *B. Hardware Configuration and Programming Languages*

This section presents concepts on how to configure the hardware, the communication connections and some programming elements. In Fig 2 are presented the components used for the application.

*Hardware Configuration* is a tool used to configure and assign parameters to the hardware of an automation project.

The following programming languages are used. The graphic programming language *Ladder Logic* (LAD) is based on the representation of circuit diagrams for programming the PLC [9]. LAD will be used to design the

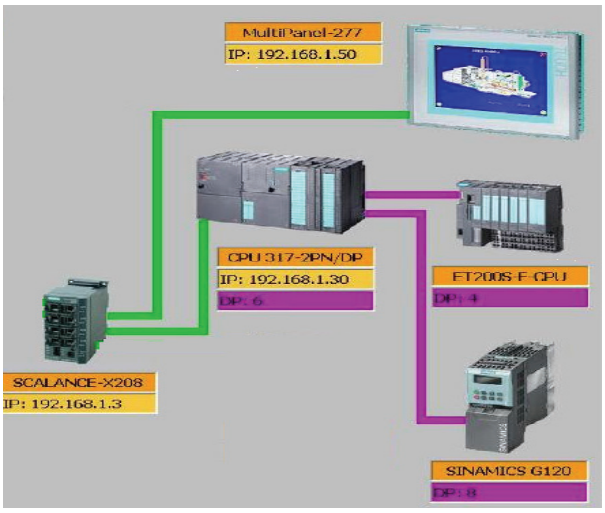


Fig. 2. Components of SCADA – TIA Democase.

control logic for the manual drive control. *S7Graph* [10] is used to implement the automatic drive control. *WinCC Flexible* is employed as SCADA platform.

At the end of this section, the hardware configuration for the application and some short examples using the programming languages will be implemented by students.

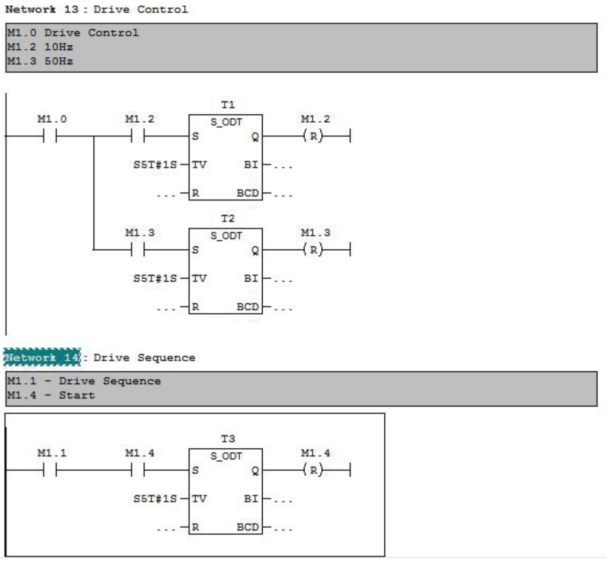
### C. *Programming and Commissioning for Frequency Inverter SINAMICS G120*

This section presents the frequency inverter SINAMICS G120, its parameters and telegrams and the software used for drive/commissioning of the inverter (STARTER).

The inverter is parameterized to be adapted to motor for optimal operation and protection. This is realized using one of the following operator units:

- Keyboard and display unit (Operator Panel) that is snapped onto the inverter;
- Software (STARTER commissioning tool) that allows the inverter to be parameterized and controlled by a PC.

Although inverters can be parameterized for very specific applications, many standard applications can be configured by means of just a few parameters [11]. The specific telegrams are understanding and developed.



```
Network 13 : Drive Control

M1.0 Drive Control
M1.2 10Hz
M1.3 50Hz
```

```
Network 14 : Drive Sequence

M1.1 - Drive Sequence
M1.4 - Start
```

Fig. 3. LAD program for inverter control.

As a workflow for the section, a LAD program (Fig. 3) will be implemented for the following functionalities: motor off, motor on, start motor at a preset frequency (25 Hz), reverse sense, initialize the system (start direct starter, motor off, frequency 0 Hz).

### D. *Manual Control for Frequency Inverter*

The aim of this section is to implement the manual drive control for the frequency inverter. The manual operating mode for the motor assumes the following:

- Start the motor at a preset frequency (10, 25 and 50 Hz);
- Stop the motor.

The three-phase induction motor is controlled by the inverter. The motor starter will protect and switch any three-phase loads. The ensemble formed by ET200S (Direct starter), inverter and motor has to be programmed. For programming the inverter, telegrams will be sent to it. In the hardware configuration for SINAMICS G120, the standard telegram 1 was selected.

Ladder logic LAD is used for programming the manual drive control [9].

### E. *Automatic Control for Frequency Inverter*

This section's main scope is the implementation of automatic control of the frequency inverter used for controlling the induction motor. The automatic control consists of a sequence of steps executed in a defined order.

A sequencer consists of a series of steps and transitions that are activated in a fixed order depending on the step enabling conditions. A sequencer is always executed starting with an initial step, or several initial steps located at any position in the sequencer. As long as the actions of a step are being executed, this step is active. If several steps are being executed at the same time, they are all active steps. An active step is exited when the transition following this step is satisfied. The next step following the satisfied transition becomes active.

At the end of a sequencer, there is a jump to any step in this sequencer or to another sequencer (this allows cyclic operation of sequencer) or a branch stop. The sequence of steps stops when the branch stop is reached [10].

In Fig. 4 are presented a few usual sequencer types in S7 Graph: linear sequencer, alternative and simultaneous sequencer and parallel sequencer.

The following sequence will be designed:
Step 1 – Run at 10Hz,
Step 2 – Hold for 10s 10Hz,
Step 3 – Run at 25Hz,
Step 4 – Hold for 10s 25 Hz,
Step 5 – Slow down to 10 Hz,
Step 6 – Hold for 10s 10Hz,
Step 7 – Ramp up to 50 Hz,
Step 8 – Hold 50Hz for 10 s,
Step 9 – Reverse rotation sense to -50 Hz,
Step 10 – Hold -50 Hz for 10 s,
Step 11 – Slow down to 0 Hz,
Step 12 – Stop.

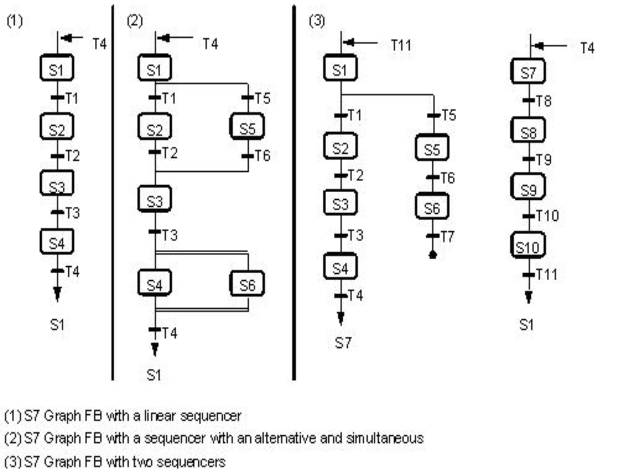

(1)
(2)
(3)
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
(1) S7 Graph FB with a linear sequencer
(2) S7 Graph FB with a sequencer with an alternative and simultaneous
(3) S7 Graph FB with two sequencers

Fig. 4. Sequencer types in S7 Graph.

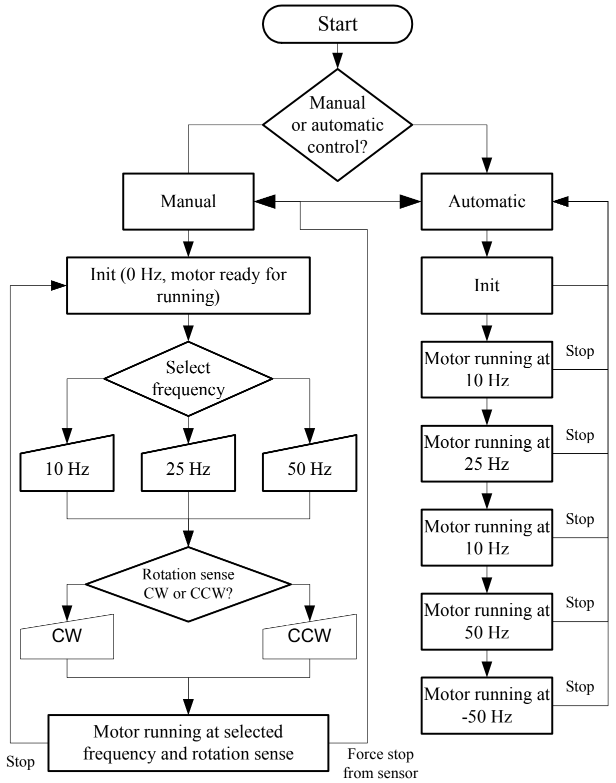


Fig. 5. Flowchart of the Manual and Automatic control modes.

### F. *Integration of Manual and Automatic Control Modes for Frequency Inverter*

This section presents the integration of the two inverter functioning modes (Fig. 5). The integration means to bring together the two control modes into one system, and ensuring that they function together as a system.

The manual control mode is the first subsystem that was implemented in section II D, and the automatic control is the other one implemented in section II E. The subsystems have to cooperate so that the system is able to deliver the over-arching functionality.

In this case, there are a series of conditions to consider:

- The switching between the two subsystems (operating modes of the inverter) is possible at any time;
- A force stop is available in each subsystem, when the inverter brings the motor in its initial state.

Only in the manual control mode, the proximity sensor is on. When the IQ-Sense module detects an object, the motor is stopped and the motor starter is disabled (contactor off). In this case, an initialization is necessary to start running again the application [12].

### G. *Introduction to SCADA Platform (WinCC Flexible)*

This section introduces WinCC Flexible - the software for programming the *human machine interface* (HMI) in SCADA applications.

The HMI provides the transparency needed between the operator, who works in environments where the processes are becoming more complex, and the machine. The HMI system represents the interface between man (operator) and process (machine/plant). There is an interface between the operator and WinCC flexible (at HMI device) and an interface between WinCC flexible and controller. The controller is the actual unit that controls the process.

An HMI system assumes the following tasks:

- *Process visualization*

The process is visualized on the HMI device. The screen on the HMI device is dynamically updated. This is based on process transitions.

- *Operator control of the process*

The operator can control the process by means of a graphical user interface (GUI), e.g., the operator can preset reference values for the controls or start a motor.

- *Displaying alarms*

Critical process states automatically trigger an alarm, for example, when the set point value is exceeded.

- *Archiving process values and alarms*

The HMI system can log alarms and process values. This feature allows to log process sequences and to retrieve previous production data.

- *Process values and alarms logging*

The HMI system can output alarms and process value reports. This allows printing out production data at the end of a shift, for example.

- *Process and machine parameter management*

The HMI system can store the parameters of processes and machines in recipes. For example, these parameters can be downloaded in one pass from HMI device to PLC to change over the product version for production [13].

A full automation solution not only involves a HMI system such as the WinCC-Flexible, but also additional components, e.g. controller, process bus and periphery.

Process tags provide the link for communication between the controller and HMI system. Without the advantage of the Totally Integrated Automation, each tag would have to be defined twice: once for the controller and once for the HMI system. The elements necessary for programming the HMI are: tags, screens (objects used on the screens), alarms, trends and special objects.

Students will create the connection between the PLC and HMI, design some screens with basic buttons, define all necessary tags and assign them to buttons.

### H. *Programming Elements in WinCC Flexible*

This section presents some special objects used in WinCC Flexible: trend view, gauge, alarms, scripting [14] and the finalization of the main application.

The *trend view* is a dynamic display object. It is meant for the graphical representation of tag values from the current process or from an archive in form of trends. The Trend view can display actual process data or data from a log continuously when it is supported by the HMI device. The *gauge* is a dynamic display object. It displays analog numerical values using a pointer.

The main tasks of the *alarm* system are:

- Visualization on the HMI: report events or states that occur in the plant or process. A state is reported as soon as it occurs.
- Reporting: Alarm events are output to a printer.
- Logging: Alarm results are saved for further editing and evaluation.

WinCC flexible provides predefined system functions for a common configuration task. They can be used to perform many tasks in Runtime and need no programming skills. Runtime *scripting* is used to solve more complex problems.

The workflow for this practical lab is the following:

- Create a script to convert the frequency value from hexadecimal to decimal, and associate the value to an internal tag.
- Insert into the manual drive control a gauge for the frequency value (Fig. 6).
- Create a screen for automatic drive control (Fig. 7).
- Create a trend view to monitor the frequency value in a separate screen (Fig. 8).
- Create two analog alarms: one for the frequency being below 25 Hz and another one for the frequency being above 45 Hz.
- Create discrete alarms to notify when the frequency is at 10 Hz, 25 Hz and 50 Hz.
- Create a screen for the alarm view (Fig. 9).
- Insert buttons in both operating modes for trend view and alarms.
- Finalize the application.

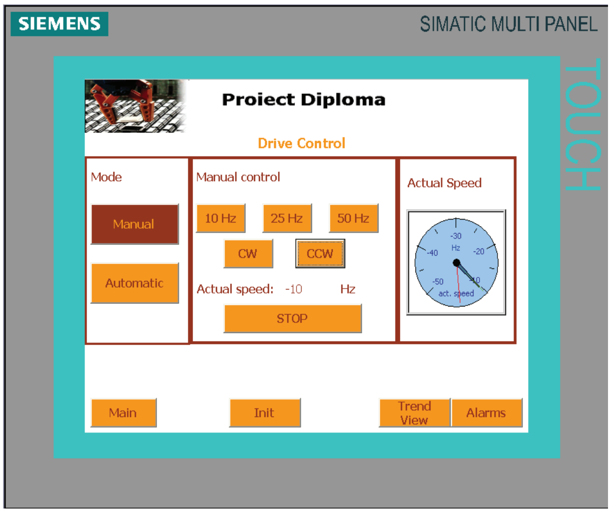


Fig. 6. Manual drive control mode.

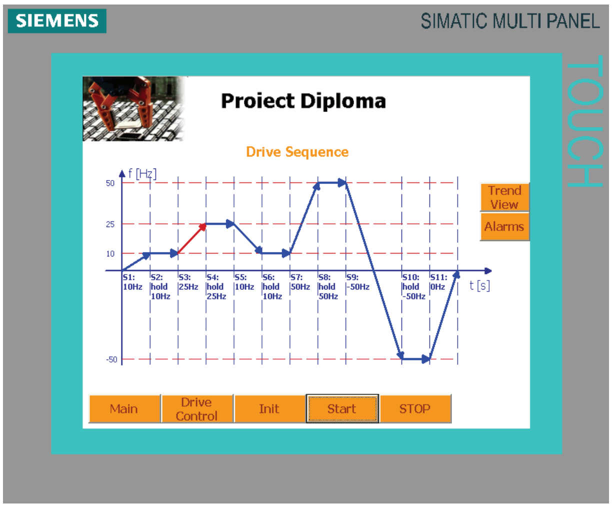
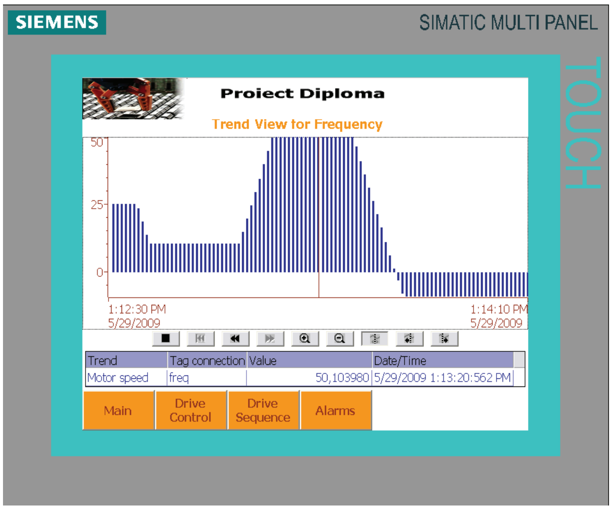


Fig. 7. Automatic drive control mode – reference sequence.



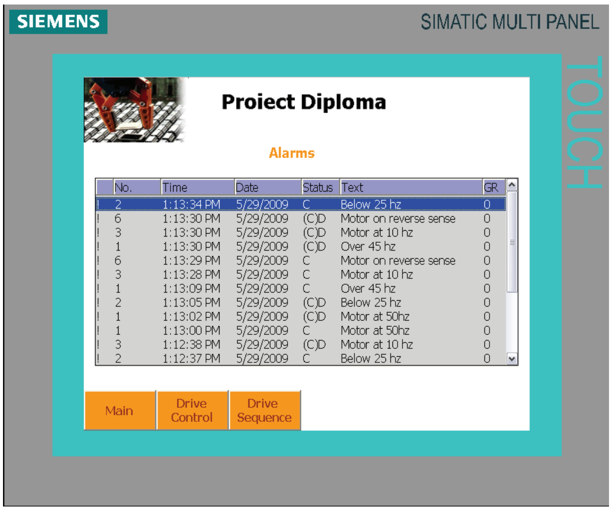
| Trend | Tag connection | Value | Date/Time |
|---|---|---|---|
| Motor speed | freq | 50,103980 | 5/29/2009 1:13:20:562 PM |

Fig. 8. Trend view for inverter frequency.

## III. Conclusions

This paper deals with theoretical and practical concepts for student education purpose in Supervisory Control and Data Acquisition (SCADA) systems. The laboratory work package is focused to gradually develop applications using an integrated automation system from Siemens - Totally Integrated Automation Democase with a distributed peripheral employing Profibus and Ethernet communications. The final application is targeted to control an induction motor with associated frequency inverter, in interlocked manual/automatic mode, with touch-screen human machine interface (HMI) to program the motor speed and to show variables, trends and alarms.

The application was divided into eight sections with specific activities, progressively developed as a good example of an automation problem solved following the basic steps necessary for an accurate industrial solution. It includes the hardware configuration and programming languages, STEP 7 standard software package for SIMATIC PLCs and WinCC Flexible SCADA platform. Laboratory works provide for student fresh perspective and competences, significant practical experience in identification, selection and programming of needed



Proiect Diploma

Alarms

| | No. | Time | Date | Status | Text | GR |
|---|---|---|---|---|---|---|
| ! | 2 | 1:13:34 PM | 5/29/2009 | C | Below 25 hz | 0 |
| ! | 6 | 1:13:30 PM | 5/29/2009 | (C)D | Motor on reverse sense | 0 |
| ! | 3 | 1:13:30 PM | 5/29/2009 | (C)D | Motor at 10 hz | 0 |
| ! | 1 | 1:13:30 PM | 5/29/2009 | (C)D | Over 45 hz | 0 |
| ! | 6 | 1:13:29 PM | 5/29/2009 | C | Motor on reverse sense | 0 |
| ! | 3 | 1:13:28 PM | 5/29/2009 | C | Motor at 10 hz | 0 |
| ! | 1 | 1:13:09 PM | 5/29/2009 | C | Over 45 hz | 0 |
| ! | 2 | 1:13:05 PM | 5/29/2009 | (C)D | Below 25 hz | 0 |
| ! | 1 | 1:13:02 PM | 5/29/2009 | (C)D | Motor at 50hz | 0 |
| ! | 1 | 1:13:00 PM | 5/29/2009 | C | Motor at 50hz | 0 |
| ! | 3 | 1:12:38 PM | 5/29/2009 | (C)D | Motor at 10 hz | 0 |
| ! | 2 | 1:12:37 PM | 5/29/2009 | C | Below 25 hz | 0 |

Fig. 9. Alarms.

to analyze and design such systems. One of the SCADA labs goals is to help students in developing skills necessary to design and analyze professional automatic control systems in process, plant and manufacturing technology engineering.

The concepts and software technologies used to develop the present application can be reused to write many other similar control/supervisory applications.

## REFERENCES

[1] M. Iacob, G.-D. Andreescu, and N. Muntean, "SCADA system for a central heating and power plant," in Proc. *5th International Symposium on Applied Computational Intelligence and Informatics SACI 2009*, Timisoara, Romania, May 2009, pp. 159–164.

[2] M. P. Groover, *Automation, Production Systems, and Computer-Integrated Manufacturing*, 3rd Ed., Pearson Education Inc., Prentice Hall, 2008.

[3] S. A. Boyer, *SCADA: Supervisory Control and Data Acquisition*, 2nd Ed., ISA, 2004.

[4] F. Adamo, F. Attivissimo, G. Cavone, and N. Giaquinto, "SCADA/HMI systems in advanced educational courses," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 1, pp. 4–10, Feb. 2007.

[1] D. Bailey and E. Wright, Practical SCADA for Industry, Newnes Elsevier, 2003.

[5] http://www.automation.siemens.com/sites-static/TIA/en/index.html

[6] *Siemens Simatic Manager Help Manual*, 2006.

[7] *TIA Democase Anhang C-Komponenten des TIA-Demokoffers.doc*, 2007.

[8] *Siemens Simatic Manager Help on LAD*, 2006.

[9] *Siemens Simatic Manager Help on S7Graph*, 2006.

[10] Siemens, *SINAMICS*

[11] *G120 CU240S and CU240E Control Units, FW 3.2Operating Instructions CU240S and CU240E Control Units, FW_3.2_enUs.pdf*, 2009.

[12] *Siemens, IQ Sense, LV20_02_IQSense.pdf*, 2004.

[13] *Siemens WinCC Flexible Help Manual*, 2006.

[14] M. S. Thomas, P. Kumar, and V. K. Chandna, "Design, development, and commissioning of a supervisory control and data acquisition (SCADA) laboratory for research and training," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1582-1588, Aug. 2004.