Communication Between TSX Micro 37-21 PLCs on Beverage Production Plant

Handy Wicaksono, Chandra Udayana, Hany Ferdinando
Electrical Engineering Department, Petra Christian University
Jl. Siwalankerto 121-131 Surabaya 60236 Indonesia
Email: handy@petra.ac.id, hany@petra.ac.id

Abstract

Programmable Logic Controller (PLC) is the most famous controller in industry. Because the process in industry is getting more and more complex than before, the usage of more than one PLC is needed and the communication between them have to be existed. This paper describes about the study of communicating two Schneider TSX Micro 37-21 PLCs in order to control the miniature of beverage production. The usual (and expensive) way to communicate them is using converter module TSX P ACC 01. Less expensive way uses original converter cable TSX PCX 1031 that connect to the ports of PLCs (TER and AUX port) which will be explained here. PL7 Pro V 4.0 software is used to program the PLCs. The communication cannot be established due to some improper addressing. The exploration is continued to the serial communication. It is found that the system uses RS 485.

Keywords: Schneider TSX Micro 37-21 PLC, PLCs communication, beverage production plant miniature.

Abstrak

Programmable logic Controller (PLC) adalah kontroler yang paling populer di dunia industri. Karena proses di industri semakin kompleks, diperlukan penggunaan lebih dari satu PLC yang dapat berkomunikasi satu sama lain. Penelitian ini menjelaskan tentang studi dalam mengkomunikasikan 2 PLC TSX Micro 37-21 dari Schneider untuk mengendalikna miniatur produksi minuman. Cara yang umum (dan mahal) ialah dengan menggunakan modul TSX P ACC 01. Cara yang lebih murah ialah menggunakan kabel converter asli TSX PCX 1031 yang menghubungkan port LPC (TER dan AUX port) yang akan dijelaskan di sini. Software PL7 Pro V 4.0 digunakan untuk memprogram PLC. Komunikasi tidak dapat dilakukan karena pengalamatan yang kurang tepat. Eksplorasi dilanjutkan kepada komunikasi serialnya. Dari percobaan ditemukan bahwa sistem ini menggunakan RS 485.

Kata kunci: PLC TSX Micro 37-21 Schneider, komunikasi antar PLC, miniatur plant produksi minuman.

Introduction

Nowadays, most of processes in industry are controlled by Programmable Logic Controller (PLC). Its ability to control the process only by programming the simple language, called ladder diagram, without any addition of hardware component, makes PLC get more and more popular than before. High flexibility and reliability are the other strengths of PLC [2].

Control System Laboratory in Electrical Engineering Department - Petra Christian University had some Schneider TSX Micro 37-21 PLCs. They have been used to control many plant on research. The example is one research that has been done by Antoni Susiono, that use it to control miniature of water level plant [11].

Because of the rapid development in industries, sometimes single PLC is not enough to control the process. When the process becomes more complex, single PLC becomes slower to control the plant. When we connect them, we have to understand about the PLC's communication and networking.

Many vendors of PLC have their own way to build PLC networking. It also happens with Schneider that produces TSX Micro 37-21 PLC which will be explained in this paper. But many times, this way makes the user spend much money. As an example to buy converter module to build a simple networking, they have to spend about US$600. For the small user who does not need more functionality of PLCs networking, this solution is too expensive. There is another way that cheaper than the first one. It utilize the TER and AUX ports of the PLC [13].

The topic about PLC networking are seldom described in the researchs. Usually they describe about PC networking, like one that explained by Irwan Sahli et al. [10].

The plant is beverage production plant. This plant is made in small size (miniature) in order to be an experiment module for the student who studies automation system with PLC. It does these basic functions: mixing two kinds of liquid from different tank, heating the mixed liquid, filling the glass with that liquid, and pooling the glass together in one place.

PLC Configuration

Schneider TSX Micro 37-21 PLC and its module

The PLC (CPU) that is used here is Schneider TSX Micro 37-21. It will use digital input/output module TSX DMZ 28 DR that has 16 digital inputs and digital outputs. It will also use analog input module TSX AEZ 801 that has 8 analog inputs. It will be programmed with PL7 software.

Schneider TSX Micro 37-21 PLCs communication configuration [13]

Figure 1 shows general way in Schneider PLC communication. It uses converter module TSX P ACC 01 to connect the Master PLC and Slave PLC. This takes an expensive way because of the existence of the converter module.
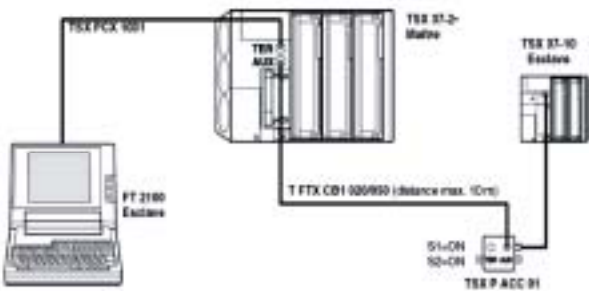

Figure 1. Configuration of PLCs communication using the converter module

The other way is make a direct (one on one) connection between two PLC, using their own port. This is cheaper way because it doesn't need any converter module. Look at figure 2.
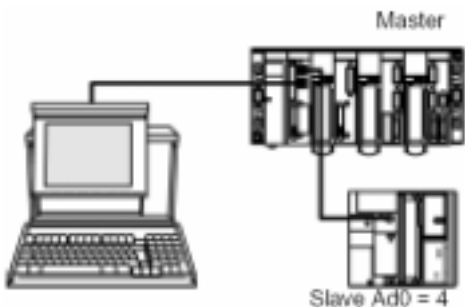

Figure 2. Configuration of PLCs communication without the converter module

Schneider TSX Micro 37-21 PLC has two ports which are called TER and AUX port. It has the same physical form, but the characteristics are different. Look at figure 3 and figure 4 to see the difference.

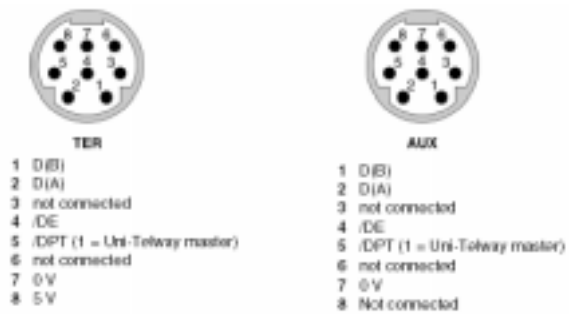Figure 3. Terminal Port on TSX Micro 37-21 PLC

Figure 4. Characteristic of TER *Port* and AUX *Port*

There are some basic instructions that used in PLCs communication. They are: READ_VAR and WRITE_VAR. READ_VAR function can be used to read value from one or some language objects, such as: internal bit, internal word, system bit, system word, constant word, structured objects (timer, monostable, counter, register, drum), etc. Its syntax is: READ_VAR(ADR#0.0.6, '%MW', 100, 10, %MW 10:10, %MW40:4) Look at table 1 to see its explanation.

Table 1. Explanation of READ_VAR syntax

Parameter	Description
ADR#0.0.6	0 : *Module*
	0 : *Channel* 0
	6 : *Sender Address* Ad0
%MW	*Type Object*
100	*Address of first object*
10	*Number of consecutive objects*
%MW10:10	*Content of reply*
%MW40:4	*Report*

WRITE_VAR function can be used to write value to one or some language objects, such as: internal bit, internal word, system bit, system word, constant word, etc. It has this syntax: WRITE_VAR (ADR#0.0.6, '%MW', 0, 50, %MW100:50, %MW 50:4)

The explanation of this syntax is almost the same with the previous one.

System Design

Software Design

First of all, the flow chart of the system will be designed here. So the process sequences of the plant can be understood. It consists of two flow charts, one for each PLC.
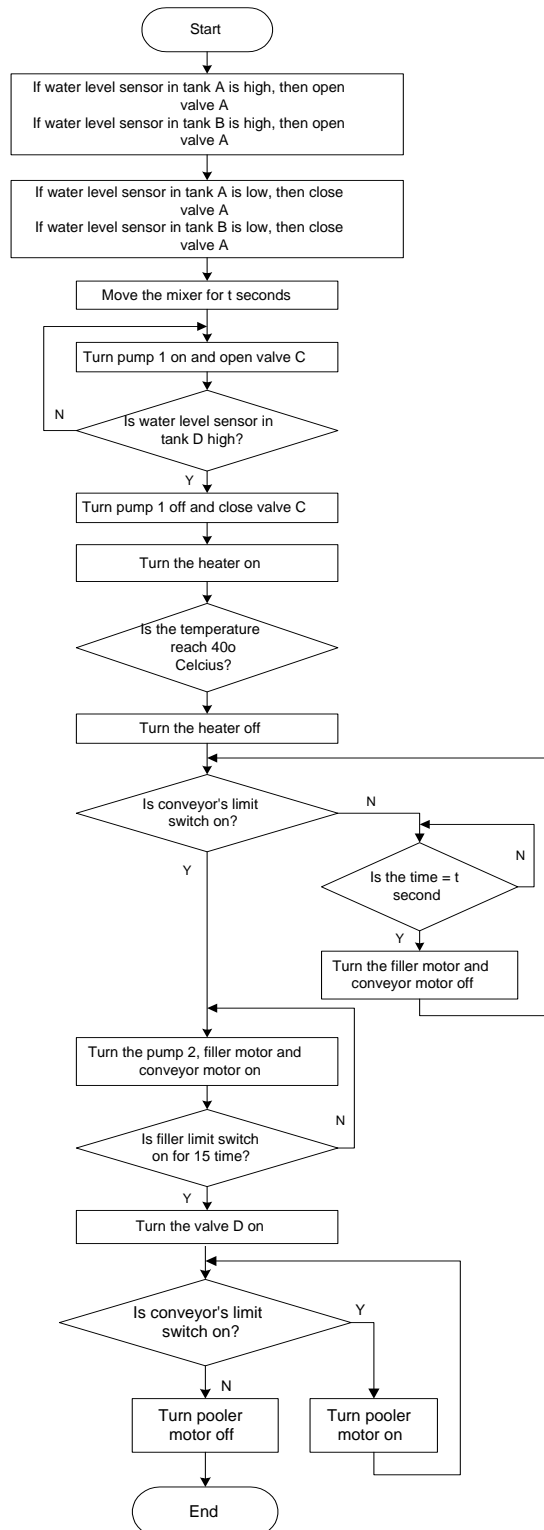

Figure 5. Flow chart of the PLC

Plant Design

Here is the general block of beverage production plant. The material A and B will be mixed in one tank. After that, it will be heated for some minutes, until it reach 40^0 Celcius. Then the liquid will be filled into the glass that moved on a conveyor. At last the glass will be gathered in a certain place. First section is controlled by PLC 1, and the second one is controlled by PLC 2. Look at figure 7.

The raw material processing plant is consist of : tank A and tank B for the different kind of liquids, tank C for mixing that liquid, tank D for heating the liquid, 2 water level sensor in tank A and tank B, mixer, heater, 2 valves and 2 pumps. Look at figure 8 to see the plant design.

After the raw material (liquid) has been processed, it will be filled into the glass that moving on a conveyor. This filler plant consist of : conveyor with AC motor as its mover, filler carousel as the heart of this filler which has DC motor as its mover, glass pooler which has DC motor as its mover, valve, and limit switches to sense the glass position. Look at figure 9 to see the plant design in the and side view.

The top view of the plant on figure 10 give clearer understanding about how the filling process happens. The glass is get into the carousel one by one (following the mechanic step of the carousel), and will be filled there.
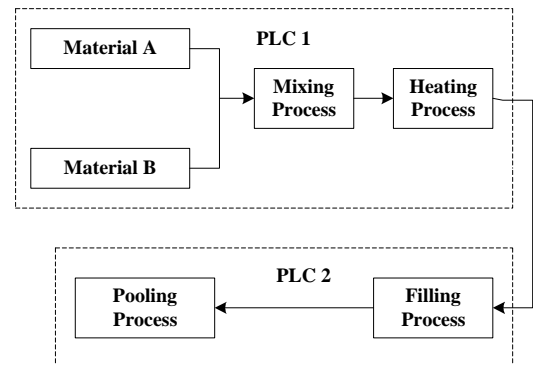

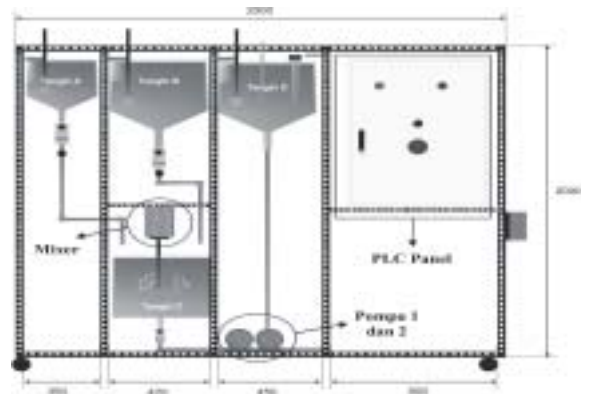
Figure 7. General block of the plant
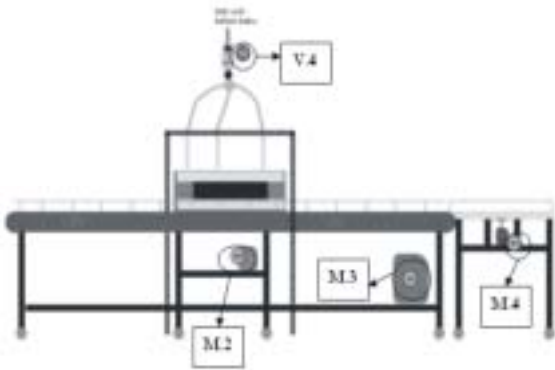

Figure 8. Raw Material Processing Plant
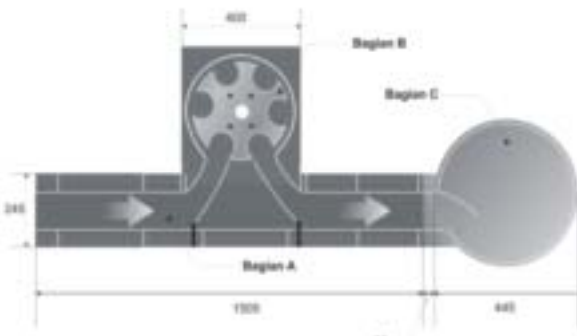
Figure 9. Side view of filler plant

Figure 10. Top view of filler plant

PLC design

1. PLC input/output design

The digital input/output devices on the plant will be handled by The digital input/output module PLC. These experiments will use two PLC for simulating how they communicate each other. Look at this digital input/output device table.

Tabel 2. Digital input/output addressing

Input/output device	PLC 1	PLC 2
AC Motor (Inverter)	Q2.10	-
Top Valve (2)	Q2.11	-
Top Valve (1)	Q2.9	-
Pooler motor	-	Q2.10
Conveyor motor	-	Q2.9
Filler motor	-	Q2.11
Filling pump	-	Q2.8
Down Valve	Q2.8	-
Filling Valve	-	Q2.7
Pump I	Q2.7	-
Heater	Q2.6	-
Start/Stop push button	I1.0	-
Limit switch 1	-	I1.1
Limit switch 2	-	I1.2
Limit switch 3	-	I1.3

And the analog input module PLC will handle four water level sensors in each tank.

2. PLCs communication design

To make a connection between PLCs, first of all, it has to be chosen which one of them will be the master PLC. So the other is a slave PLC. The definition of its status will be done in PL7 software, in hardware configuration section. Look at figure 11 to see the window that appear.

In A section, we choose the COMM section, then in the B section we will choose whether the PLC is the master PLC or the slave one. At last, in C section it will be chosen how many slaves that will connect to the master PLC (if the PLC is the master), or the address of the server (if the PLC is the slave).
Only the master PLC can read data from the slave one and it can also write data to the slave PLC. So the READ_VAR and WRITE_VAR only can used by master PLC. The slave PLC can not do anything in communication. Here is the example ladder diagram.
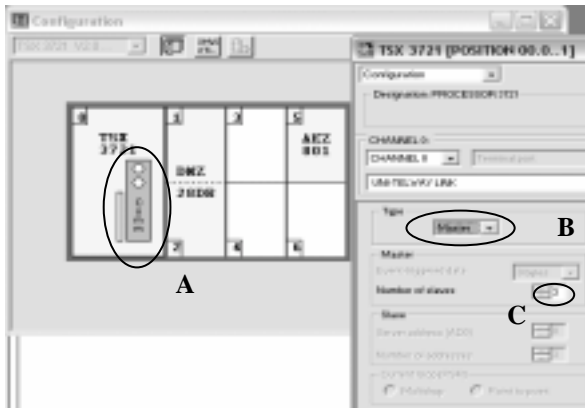

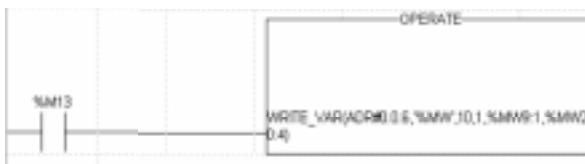Figure 11. Example of hardware configuration for master PLC with PL 7 software

Figure 12. Ladder diagram example in master PLC

Experiments

PLCs Communication Experiments

Experiments on PLCs communication is done by connecting two Schneider TSX Micro 37-21 PLCs. As described on this paper's abstract, the usual connection between them by using converter module TSX P ACC 01 won't be covered here, because there is another way which is less expensive then the former. So the communication is made by connect them directly on their own port using original

converter cable TSX PCX 1031. The first PLC use AUX port, and the other use TER port. The connection is made after the ladder diagram is downloaded from PC (Personal Computer) to each PLC.

After we change the PLC into the RUN state, the warning box in figure 13 will appear. Although there is an error in PLCs communication, the first PLC still can do its job well. The sequence of the process in first PLC is running well until the second PLC's turn to handle the process. The second PLC can not do its job well.
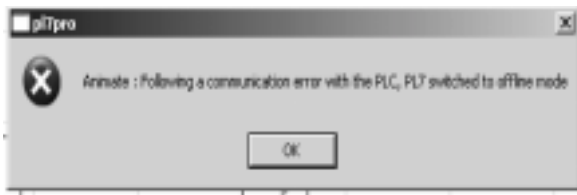

Figure 13. The first warning box

This experiment is failed because address on second PLC (slave PLC) can not be changed to ADR#0.06, because when it receives the program from other device (example: personal computer) the address automatically become : SYS. That's why the first PLC can not send data to the second one. The first PLC always send data by using ADR#0.0.6, like the explanation on PLC's manual which is stated that to reach the slave PLC by UNITELWAY protocol, it has address form module.channel.slave number.

Because of there is error, the PLC should not be transferred the new program for 1 minute and 30 seconds. It is explained on the second warning box on the figure 14.
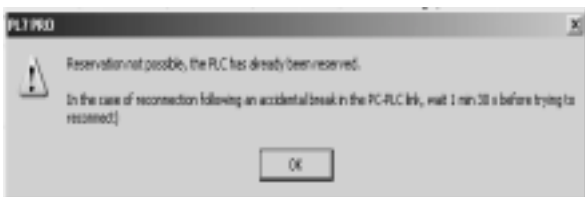

Figure 14. The second warning box

The experiment has be done by changing the address of slave PLC with form module.channel.slave number, but when the program will be transferred from PC to PLC then it failed again. And the warning message on Figure 15 will appear.
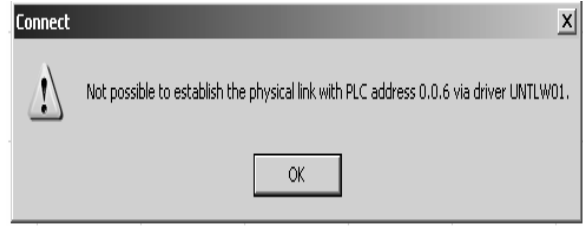

Figure 15. The third warning box

The next experiment is by changing the selection of PLC ports that used in the communication. The first PLC used TER port and the second used same port. There is no warning message like the previous experiments, but the result is just the same. After the sequences on the first PLC have been executed successfully, they could not run well on the second PLC.

Signal Monitoring experiments on TSX PCX 1031

The PC is connected with PLC by using its serial port. It means that RS 232 serial communication has been used here. There are only three pin needed to make this kind of communication work: Tx (transfer pin), Rx (receive pin) and Ground pin.

The Schneider TSX Micro 37-21 PLC has a special module (which is included in communication cable) that called TSX PCX 1031. This device is connected with the TER port on PLC, and it got its supply voltage from PLC. It can be said that PLC do not use RS 232 like the PC, but it still use serial communication because it serially connected with PC's serial port.

Then to find what kind of serial communication that existed on PLC, this experiment is done. When the PLC is running, the signal between D(A) and D(B) is captured in oscilloscope. The result is appeared on Figure 16 and 17.

From the figures, it can be analyzed that there is differential voltage between D(A) pin and D(B) pin. And the amount of the voltage is exactly the same (its about 5 V). It means that D(A) pin use D(B) pin as reference and the same thing happen with D(B) pin. So that it can be concluded that the kind of serial communication is multipoint. Because the number of the main wire is two, it should be RS 485 serial communication.
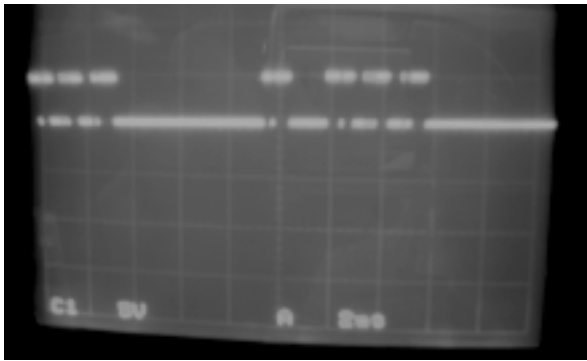
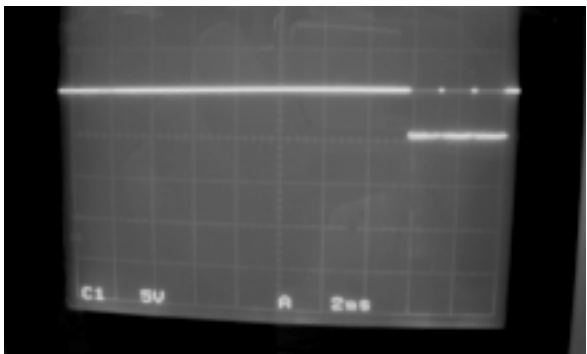Figure 16. Signal between D(A) and D(B) on TSX PCX 1031

Figure 17. Signal between D(B) and D(A) on TSX PCX 1031

Conclusion

1. Direct communication between Schneider TSX Micro 37-21 PLCs by using their own ports still unsuccessful, because of improper addressing. In transferring program between PC & PLC use SYS address, while between first PLC and the second one use ADR # 0.0.6

2. PLC – PC communication needs a converter because PC uses RS 232 serial communication, while PLC uses RS 485 (multipoint) serial communication. Multipoint serial communication used by PLC produce differential voltage (peak to peak): 5 Volt.

Reference

[1] B&B Electronics, *RS-422 and RS-485 Application Note*, Ottawa, 1997 <http://www.bb-elect.com>

[2] Bolton, William. *Programmable Logic Controller (PLC) Sebuah Pengantar (3rd Ed.)*. Jakarta: Erlangga, 2003

[3] I Made Arya Djoni. *Pompa & Compressor TM 1532*. Surabaya: Jurusan Teknik Mesin FTI-ITS, 1996

[4] *Instruction Manual Fuji General Purpose Inverter FVR-E9S Series*. Japan: Fuji Electric Co.,Ltd, 1997.

[5] Kissel, Thomas E. *Industrial Electronics*. Singapore: Prentice Hall International Inc, 1997.

[6] Modicon. *Modicon Modbus Protocol Reference Guide*. CD-ROM. Modicon Modbus Protocol Reference Guide. Paris, 2003.

[7] New York Institute of Technology. *RS 232 Serial Interface*. New York, 2005. <http://iris.nyit.edu/~spatanga/RS-232.doc>.

[8] *PL7 Micro/Junior/Pro Communication applications-volume 1*. CD-ROM. PL7 Micro/Junior/Pro Communication applications. Paris: Schneider, 2002.

[9] Stallings,William. *Dasar-dasar Komunikasi Data*. Trans. Thamir Abdul Hafedh Al-Hamdany. Jakarta : Salemba Teknika, 2000.

[10] Sahli, Irwan. Kristanto, Irwan. Chandra, Tony. Perancangan RS 232 to RS 485 Converter Sistem Network Multidrop. Surabaya : Jurnal Teknik Elektro - Universitas Kristen Petra, Vol. 1No. 1, Maret 2001

[11] Susiono, Antoni. *Aplikasi SCADA System pada Miniatur Water Lavel Control*. Surabaya: Tugas Akhir-Jurusan Teknik Elektro-Universitas Kristen Petra, 2005.

[12] Teodore,Wildi. *Electrical Machines, Drives, and Power System*. 6th Ed. New Jersey: Prentice Hall, 2005.

[13] *TSX Micro PLCs TSX 3705/3708/3710/3720 Implementation manual – volume 1*. CD-ROM. TSX Micro PLCs TSX 3705/3708/3710/3720 Implementation manual. Paris: Schneider, 2002.