UNIVERSITY OF VAASA

FACULTY OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND AUTOMATION TECHNOLOGY

Onyebuchi Dalbert Zimuzochukwu

Z109554

IITS2203-5 Fieldbuses and Internet (I-IT-4N)

Laboratory Report 2: Fieldbus application- Vacon driven by Profibus

Instructor & supervisor: Dr. Ali Altowati

17.11.2017

Vaasa, Finland.

PURPOSE: Controlling a motor using utilizing Vacon frequency converter connected to a PLC.

THEORY:

The Vacon frequency converter is a device along with penal and a built-in motor and fan. It is controlled by the PLC.

It is used to control the electrical equipment by converting the input frequency.

We seek to perform the following controlling tasks:

- Turning the fan on and off
- Speed control
- Changing the direction of rotation

S7-300 a Siemens PLC Product is used for this purpose.

Our task is to connect the PLC to the system and to configure it, make physical connections between different hardware devices and logical programming to operate the devices. In this specific task a Vacon frequency converter is being used.

LABORATORY SETUP:

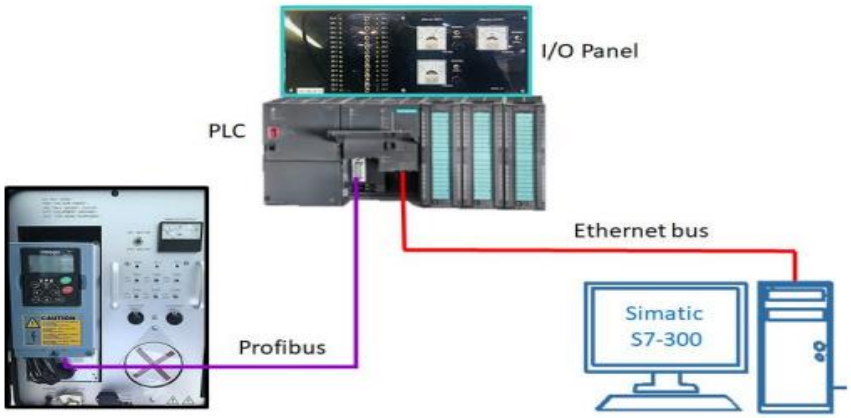


Figure 1: Connection between the PLC, PC and Frequency converter

PROCEDURE:

1. First of all we need to configure our hardware. We connected our PLC with the PC by using Ethernet cable.
2. Simatic S7 software is used to make a connection on software level of a PLC with the PC.
3. By staring a new project, PLC (S7-300) is added and then hardware configuration is done by setting the IP address.
4. Then after selecting rack, add CPU with respect to its model number.
5. Then add other digital and analog I/O module with respect to their model number.
6. We need to control Vacon frequency convertor by Profibus, so Profibus is configured.
7. Now configure Vacon frequency convertor from the folder "Additional Field Devices" with the Profibus by using its slave address which was 6 in this specific case. Messaging type "PPO3" was used.
8. Drag that type to the row number "0".
9. Activate all modules and then after saving & compile, download these entire configurations to the CPU.
12 Connect the Profibus in between the PLC and Vacon frequency convertor and check the status of PLC BF led. If it's not blinking, then we have established our connections successfully.
13 Also checked Profibus status on the screen of frequency converter. The result should be "2" which means Master-Slave connection is established.

In the second step, we write a PLC program to control the Frequency convertor.

14 Use "MOVE" block by assigning it the input and output values to "run the drive".
15 Add a second move block to "stop the device".
16 To control the speed, by analog potentiometer we created the scaling.
17 Added "DIV_I","MUL_I" and "SUB_I" blocks to adjust the range of speed.
18 Compile and save the program, then downloaded to the PLC.

The program and other configuration can be seen in the real program image below.

FIGURE 2: Configuration

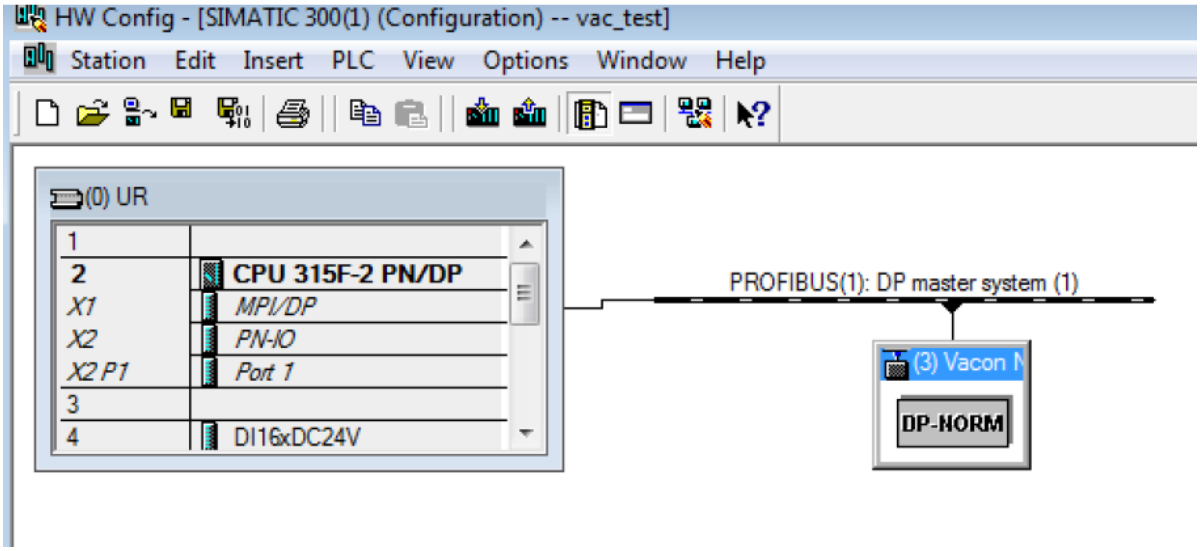
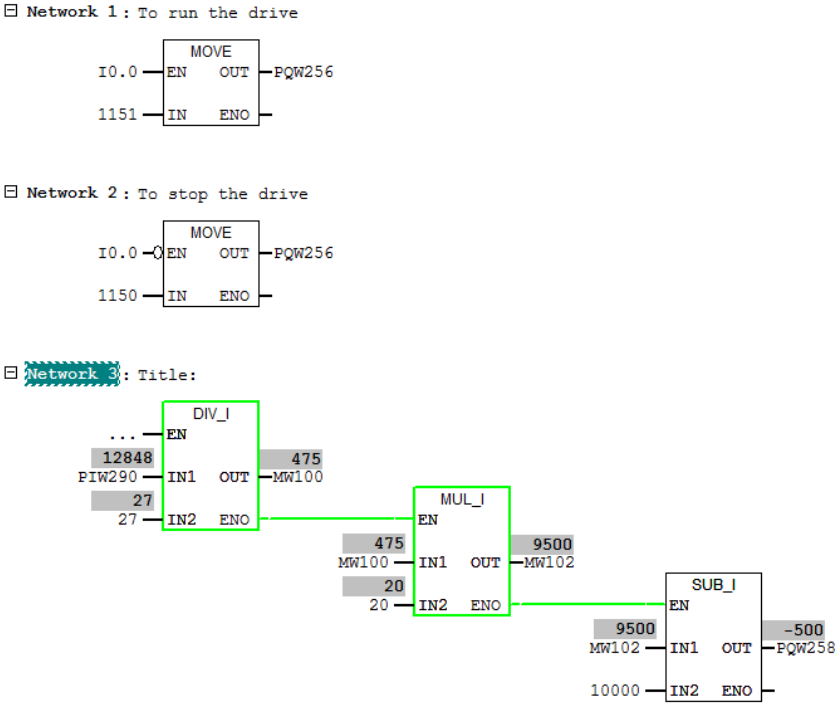
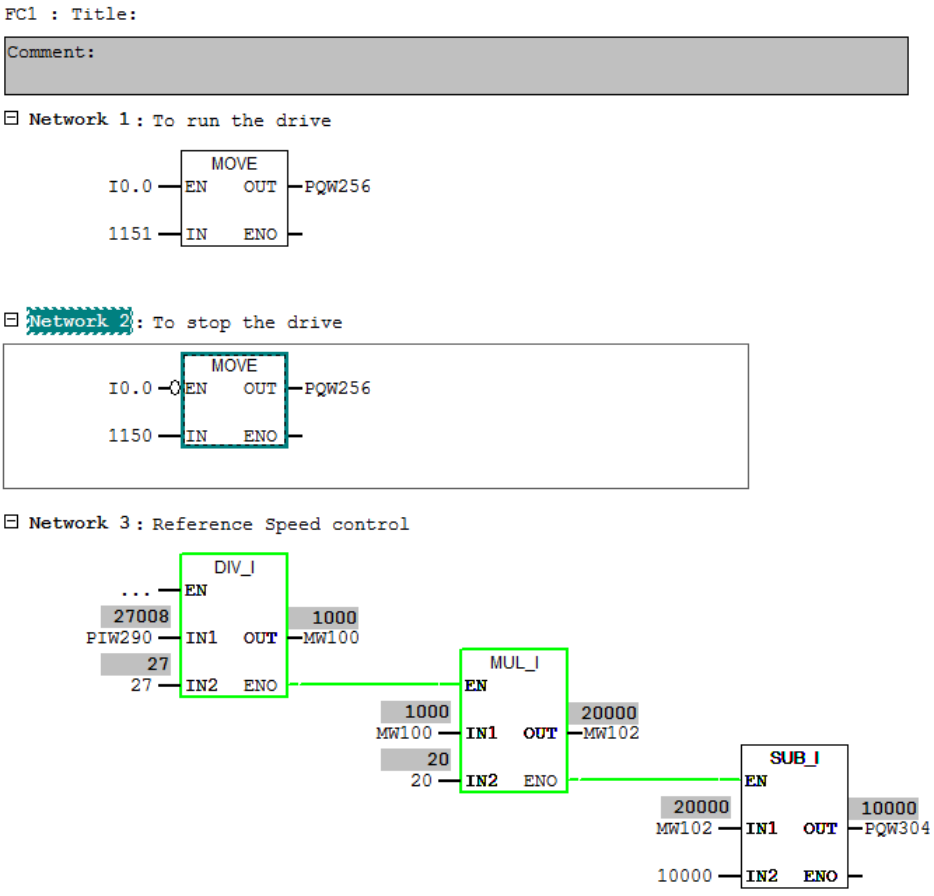


FIGURE 3: Application Program



. **SECOND SPEED TEST: Notice a difference in parameters**



OBSERVATION:

The speed of the fan in the VACON Frequency Converter changes when we change the potentiometer scale. Changing the potentiometer direction also changes the rotation direction of the motor drive. When it is on the positive side the fan motor would rotate clockwise. Changing the direction of potentiometer, makes the motor rotate in anti-clockwise direction.

Frequency Change can be seen on the frequency converter screen. The motor speed can also be controlled by changing the input parameters in the program.

Other questions:

Answer with your own words, what good is in the Profidrive-profile.

For normal data communications, there is data transparency between both devices connected as they do not know the application of the data being transferred. Each device is isolated from the knowledge of other devices in the system.

With the Profidrive-profile, some advantages over normal data communications exists. For communication, we have an additional feature that enables us can define devices. Device definition is performed at the application level.

It is termed as "Profile" and its combination along with Profibus is called Profidrive-Profile.

It works upon Master-slave principle. The Master issues a command (with control data) while the Slave responses immediately by sending back its present status. This profile contains different specifications for speed control and positioning.

How the communication bus cable should be grounded?

Grounding is a process in which some part of the electrical circuit is linked with zero potential point. It ensures personal safety and generally improves the circuit work,

There are three methods of grounding for the Bus cable shield.

i. Directly to the frame of frequency converter
ii. To the frame of frequency converter through an RC filter
iii. Clamping the cable directly to the converter frame

But as recommendation it's better to ground the communication bus cable by using the third method in the list.

Grounding by clamping the cable to the converter frame.

It's the most effective manner of grounding in which the position of jumper X1 is no more important.

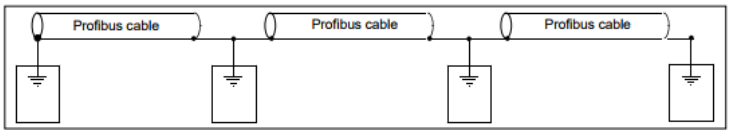


Figure 2*:* Grounding by clamping the cable to the converter frame

i. First, we will strip the Profibus up to 5cm and will cut off the grey cable shield.
ii. Leave no more than 1 cm of the red and green data cable outside the terminal block and strip the data cables at about 0.5 cm to fit in the terminals.
iii. Insert the red and green data cables of both Profibus cables into terminals #3 (red) and #4 (green)
iv. Strip the Profibus cable at such a distance from the terminal that you can fix it to the frame with the grounding clamp

How will the PLC application reach the drive?

The profile defines 5 different Control / Status data pair. Each pair carries a slightly different data. In this specific application we will use PPO3.

It is the simplest type which carries.

Command:

- CW, Control Word – 2 bytes, 16 bits
- REF, Reference – Frequency reference – 2 bytes, 16 bits

Response:

- SW, Status Word – 2 bytes, 16 bits
- ACT, Actual, – Actual Frequency – 2 bytes, 16 bits

In the standard the Frequency Reference and Actual value are scaled:

-10 000 – 10 000 = - FreqMaxLimit - FreqMaxLimit

Is there difference, how the PLC sees the drive data and the I/O-data? (Based on, what you see in the HW-configuration) ---------- (Yes/No)

YES.

Data Transfer within PLC modules, entails moving data from one memory location to another. The devices are not aware of this data (Transparency).

But in the case of drive data, the devices are aware of the location from which they are receiving the data. Because the drive data communication works upon Master-Slave Configuration principle where one device works as master and the other devices works as slaves according to instruction of the device in Master configuration.

Briefly explain the accessing process and I/O data areas in PLC?

Normally PLC stores all the information in its memory registers. So instead of getting direct access to the I/O modules, programmer directly accesses those memory areas by using the input and output operands. These input and output operands are “I” for input along with the memory address and for output it is “Q” along with its memory address. I.e. “I0.0”, “I0.1” for the input and “Q0.0” and “Q0.1” for output. This configuration is different for different PLC modules.

This specific area is called “the Process image of the inputs (PII) “and “the Process image of the outputs (PIQ)”.

PII and PIQ always reflect the specific values which are loaded on any time in any specific process. These values are keep updating during the process. So, CPU contain different process image according to their memory.

Figure 3 shows the different notation with respect to memory. Address area is shown there in the table which show the process image (I/O) inputs and process image (I/O) outputs.

It also shows how to access the units of different sizes. We always chose the unit according to our approximate data size. Normally digital I/P and O/P are stored by using simple bit addressing which is IB (input byte) but while we are dealing with a little larger size of the data we need to select IW (input word) or ID (input double-word). Same logic is used for output.

Address Area	Access via Units of Following Size	S7 Notation (IEC)
Process image (I/O) inputs	Input byte	IB
	Input word	IW
	Input double word	ID
Process image (I/O) outputs	Output byte	QB
	Output word	QW
	Output double word	QD

Figure 3

In simple words, if we need to handle digital data, then we can handle it by using IB (input Byte) but if we need to handle any kind of analog data like input, related to analog potentiometer and output, related to the volt meter so we have to chose bigger notation like IW or ID. The reason behind it that analog data needs more memory than digital data.

The process image:

It works in following steps.

- Process CPU operating system
- Writing of states from the OB1 process image of the output, to the output of the module.
- Reading of module input states
- Processing of the user program in OB1
- Repeat the process.

Peripheral Address:

These addresses are not included in to the process image. The actual values are immediately read and written to physical I/O from user programs.

They are used to handle the values which are coming directly from any external I/P and need to write directly to the output. Notation of peripheral Address is shown in the figure 4.

Address Area	Access via Units of Following Size	S7 Notation (IEC)
Peripheral (I/O) area: inputs	Peripheral input byte	PIB
	Peripheral Input word	PIW
	Peripheral Input double word	PID
Peripheral (I/O) area outputs	Peripheral Output byte	PQB
	Peripheral Output word	PQW
	Peripheral Output double word	PQD

Figure 4

The whole process is shown in the figure 5.

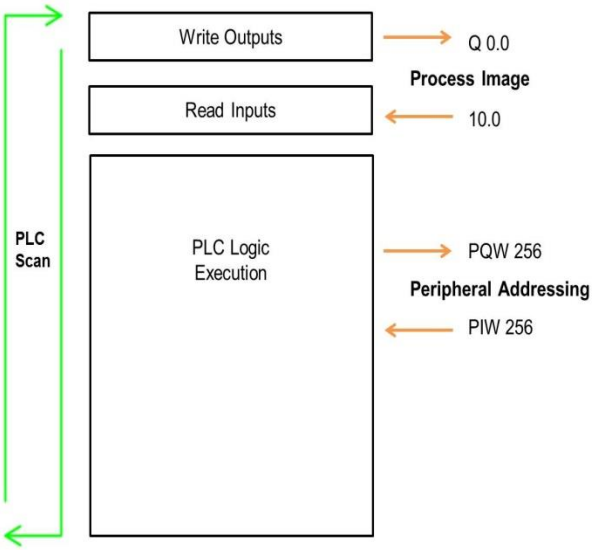


Figure 5: Accessing Process