

Ethernet Control AC Motor via PLC Using LabVIEW

Nader N. Barsoum, Pin Rui Chin

*Department of Electrical and Computer Engineering, Curtin University Sarawak Campus,
Miri, Malaysia*

E-mail: nader.b@curtin.edu.my, chinpinrui@yahoo.com

Received July 5, 2011; revised July 26, 2011; accepted September 3, 2011

Abstract

Remotely control applications over a wide area had been commonly used in the industries today. One of the common applications requires remote control and monitoring is inverter fed induction drive system. Drive system has various types of controller, in order to perform some actions such as control the speed, forward and reverse turning direction of the motor. This approach can be done by Programmable Logic Controller (PLC), and with the rise of the technology, Ethernet module will be used in order to achieve the remote control system. Plus the PLC today can be controlled not only using its original software, but 3rd party software as well, such as LabVIEW. LabVIEW is a human machine interfaces design software that is user friendly. It can be easily communicate with different hardware.

Keywords: LabVIEW, PLC, Ethernet, Induction Motor, OPC Servers, Inverter

1. Introduction

In the past, engineers had been designing the engineering systems that require a lot of hardwares. It is merely impossible to design distance control of the system as more hardwares and wiring were needed. In addition, if engineers wish to improve the design, all the irrelevant hardwares need to be scrap away, which is not sustainable.

With the rise of the technology, programmable logic controller (PLC) have eased the engineering design and lessen materials required, it is because the entire design is implemented in software programming paradigm. PLC had been commonly used in the industry, including controlling induction motor inverter fed variable drive system. Design distance control machinery is now possible, even by using Ethernet as the communication device between the computer and the PLC [1].

Apart of design the program structure by its own proprietary software, the convenience part of PLC is the accessibility and controllability by other softwares. Note that such softwares must have driver utility of the particular PLC. Therefore engineers can use LabVIEW [2], which has various types of industrial applications which are in virtual instrument (VI) instead of the real and heavy instrument, to control the PLC.

2. Project Design Configuration & Preparation

2.1. NI OPC Server

NI OPC Server has OMRON FINS Ethernet driver that allow the communication between OMRON CJ1M-CPU11-ETN21 PLC with LabVIEW. OMRON originally supplies their customers with FINS gateway, interfacing software that communicates with the PLC and its proprietary software, OMRON CX-Programmer over the Ethernet network [3].

With the OMRON FINS Ethernet driver in NI OPC, users can setup the server by just a few simple setups and create variable tags that can be linked directly to the PLC's registers. These tags are named as OPC tags. The NI OPC Servers also have NI OPC Quick Client that enable users to monitor the status of the PLC in real-time. [3].

As long as the OPC tags had been created, the communication between the LabVIEW and PLC had been simplified as the driver can automatically apply the relevant FINS commands provided the tags are correctly configured [3]. Meanwhile in LabVIEW, the program can be design by using Shared Variables which is link to the OPC tags.

2.2. LabVIEW

LabVIEW is the acronym for Laboratory Virtual Instrumentation Engineering Workbench and is a graphical development environment for generating flexible and scalable design, control, and test applications rapidly at minimal cost. With LabVIEW, engineers and scientists are able to interface with real-world signals, analyse data for meaningful information, and share results through intuitive displays, reports, and the Web. Regardless of programming experience, LabVIEW makes development fast and easy for all users [3].

The programming style used in LabVIEW is G programming, which is abbreviation for graphical programming. It is also known as dataflow programming as it is depending on the structure of the graphical block diagram to execute the user-designed program. Compared to text-based programming, LabVIEW is user friendly as the user can design the program by simply arrange and wiring the relevant icons together [3]. LabVIEW programs are named as virtual instruments, or VI, because their appearance and operation mimic the physical instruments, such as oscilloscope and multimeters [4]. Similar to other conventional programming, LabVIEW has standard features such as looping structures, data structures, event-handling, object-oriented programming. LabVIEW also has an extensive library of math functions similar to MATLAB libraries and also

formula nodes that allow text-based programming for certain sections of the code that require complex logical structures. Besides that, LabVIEW also has networking library functions that can easily allow users to reference [3].

Compared to other softwares like Microsoft Visual Basic, LabVIEW is a better option as it comes together with a library of functions included Shared Variables Project Library, which is bound to the OPC tags, that allow server and client communication by connecting relevant icons with the Shared Variables. If Microsoft Visual Basic (VB) is used, the OMRON FINS Ethernet driver must be developed using the MS Comm function and this would require more time to develop the code [3].

LabVIEW has front-end interface applications that allow user to design and then use for controlling systems. In general LabVIEW has three main elements: the front panel, the block diagram and the connector panel. The front panel allows the user to build the controls and indicators. The controls are including knobs, push buttons, dials, and other input mechanism. Indicators are graphs, LEDs, and other output displays. Meanwhile, the block diagram let user to add code using VIs and structures to control the front panel objects. The connector panel allows user to represent a single VI as a sub VI icon that can be called in another VI. The elements are illustrated in **Figure 1**.

Shared variable is a library function variable that al-

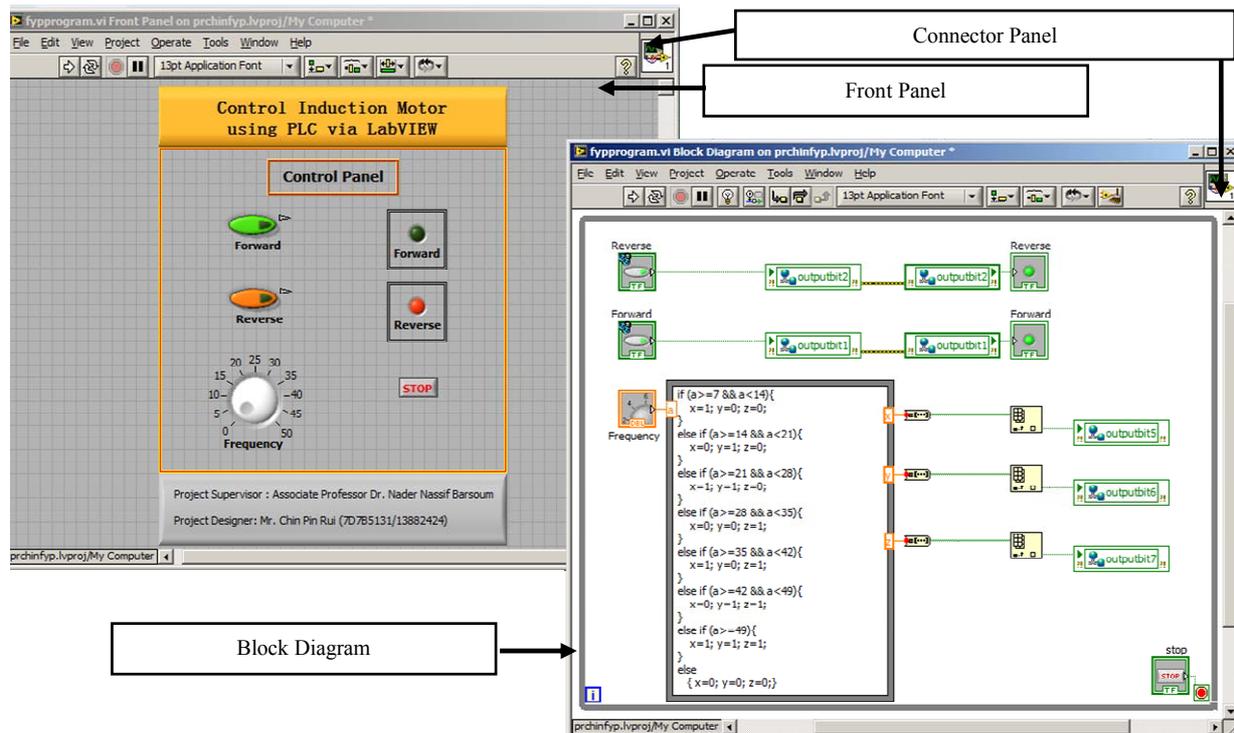


Figure 1. Three main elements of LabVIEW software.

lows sharing of data between applications or different data sources across a network. There are many existing data sharing method in LabVIEW, such as UDP/TCP, LabVIEW queues, and Real-Time FIFO. Compared to Datasocket Communication, using shared variable is user friendly as the configuration can be done by simple setup in the library instead of writing URL and perform more wiring in Datasocket Communication [3,5].

2.3. Programmable Logic Controller

The PLC used for the implementation is OMRON CJ series. There are 4 units used in this PLC, namely Power Supply Unit (CJ1W-PA202), CPU Unit with Ethernet function (CJ1M-CPU11-ETN21), Basic Input Unit (CJ1W-ID211), and Basic Output Unit (CJ1W-OC211).

There is an End Cover for the PLC unit. All these units can be connected by assembled them together and lock the sliders by moving them towards the back of the units. The End Cover must be connected on the far right hand side of the PLC. Otherwise fatal error will occur [6].

Figure 2 demonstrates the arrangement of the PLC units.

In order to let the PLC to operate through the Ethernet, the PLC must be given an IP address with a destination node number (DA1) that is not a duplicate of DA1s of other IP addresses in the network. The destination node number is also known as the last octet of an IP address.

The PLC node number is configured by turning the dials on the CJ1W-ETN21 module as illustrated in **Figure 3** the PLC must also be given a unit number in the network besides a destination node address (DNA) [3].

Node No $\times 16^1$ is the most significant bit (MSB) of the node address whereas $\times 16^0$ is the least significant bit (LSB) of the node address. IP addresses are in decimal form therefore any reference made to the DA1 of the PLC Ethernet module in software configurations must be in decimals. To convert any hexadecimal number to decimal, the following example can be applied [3]:

$$\begin{array}{r}
 \text{MSB LSB} \\
 \begin{array}{c}
 \frac{2}{\quad} \quad \frac{E}{\quad} \\
 \downarrow \quad \downarrow \\
 14 \times 16^0 \\
 + 2 \times 16^1 \\
 \hline
 46
 \end{array}
 \end{array}$$

The PLC is configured with the IP address 10.1.136.46, for the implementation described in this paper. The router is for personal home network. Therefore the IP addresses associated with the devices in the network is classified under Class C.

2.4. Variable Frequency Drive

OMRON SYSDRIVE 3G3MV-A2007 inverter is a variable frequency drive can be used to alter the frequency of the electrical power supplied to the motor so in order to change the motor speed [6].

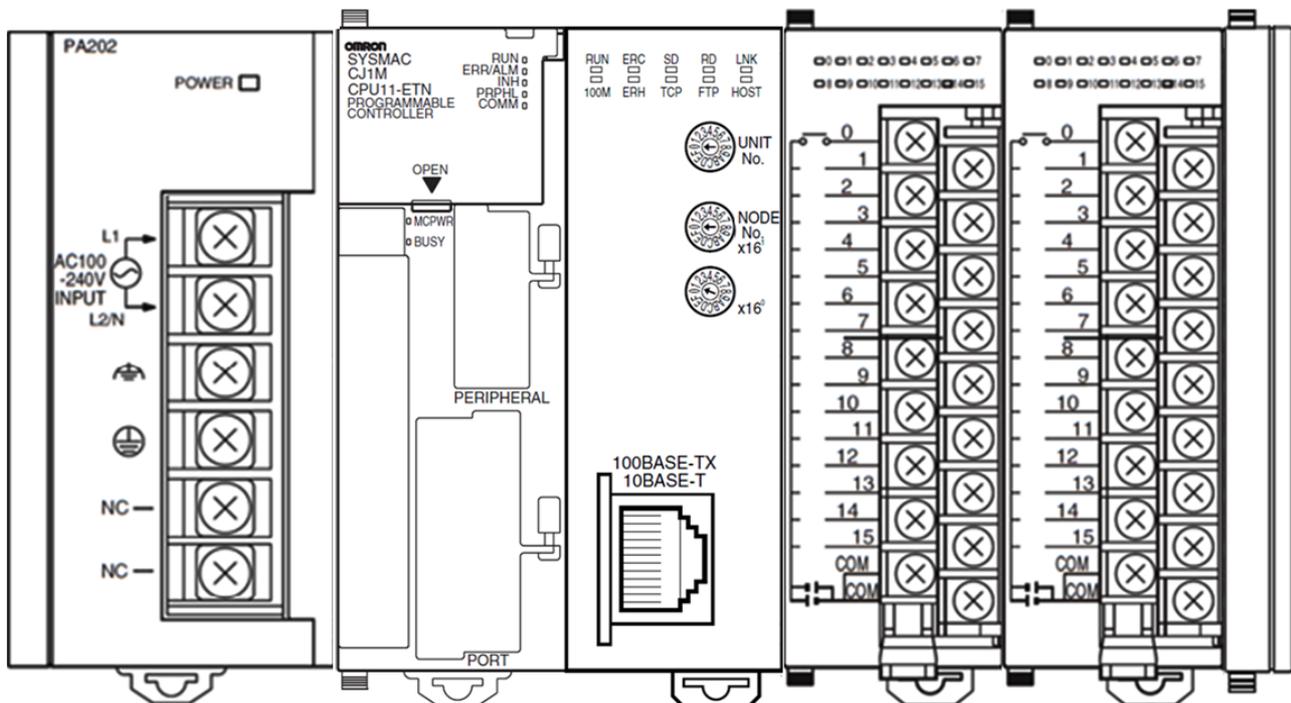


Figure 2. Connection for PLC units in this project [7,8].

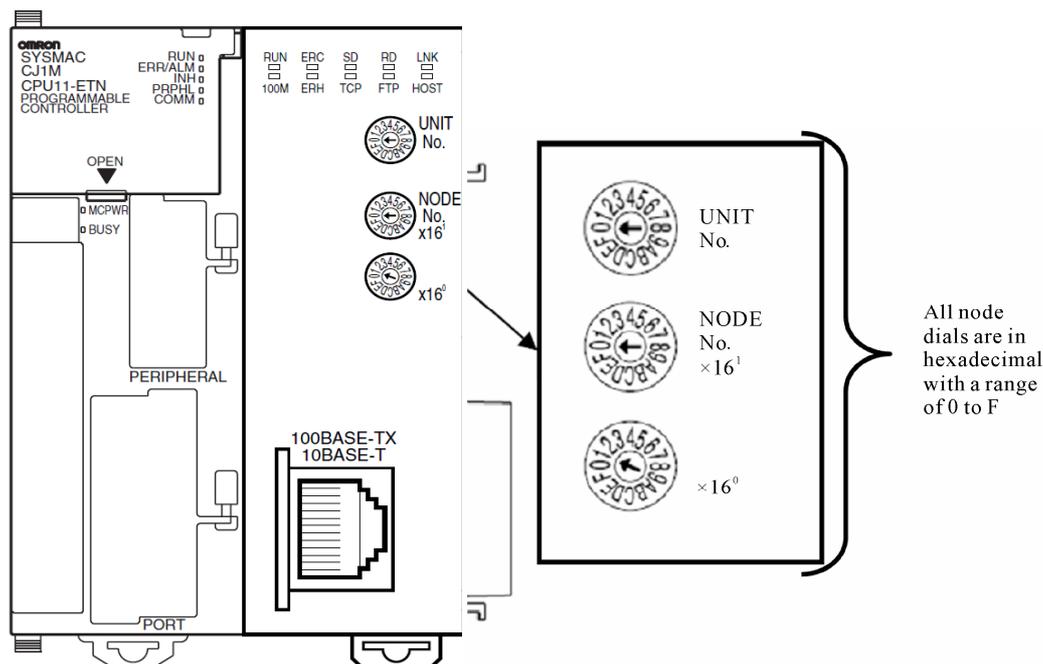


Figure 3. CJ1W-CPU11-ETN21 hardware configuration.

The SYSDRIVE 3G3MV-A2007 inverter is suitable for a variety of applications as it incorporates many convenient controls and I/O functions that are easy-to-use as well as open loop vector control function. The advantages of vector control function are that it ensures a torque output that is 150% of the rated motor torque at an output frequency of 1 Hz, allows powerful revolution at low frequencies and restrains the revolution fluctuation caused by the load [6].

The list of function parameters of the 3G3MV inverter need to be configured in order to control the 3 phase squirrel cage induction motor effectively.

3. Implementation

3.1. Implementation Process

The Ethernet control systems presented in this paper is to control squirrel cage three phase induction motor. In order to achieve the objectives, the establishment of the communication between PLC and LabVIEW is crucial as LabVIEW is 3rd party software instead of using the software implemented in the PLC itself. Thus, the implementation used LabVIEW to perform the start and stop operation of the motor, either in forward or reverse direction, and varying the speed by changing the frequency of the motor. However, this system is not a supervisory control and data acquisition (SCADA) since there is no practical data measurement acquire from the actual output of the motor.

The system has three-layer network architecture illustrated in Figure 4.

As illustrated in Figure 5, user gets the authority to control through the host computer, which is a laptop. Then the input data by the user will convert into Boolean data and send to CJ1M-CPU11-ETN21 Programmable Logic Controller through Ethernet cable and router. Once the Boolean data have been processed by the PLC, the relevant address in the basic output of the PLC will be turned on. This process allow the 3G3MV Inverter/ Variable Frequency Drive to operate the Three Phase Squirrel Cage Induction Motor according to the input data given by the user. In addition, 3G3MV Inverter/ Variable Frequency Drive also serve as inverter between the power supply unit and the motor as the input power is single phase power, while the Squirrel Cage Induction Motor is operate in three phase power.

3.2. Implementation of VI Design

The objective of the VI program in this paper is to allow user to make the decision of the start and stop operation of the motor, either in forward or reverse direction, and varying the speed by changing the frequency of the motor, by perform two simple step. Firstly select the turning direction by press the push button in the VI Front Panel, which is either forward or reverse. Secondly, vary the speed by turning the Frequency knob to the value of the frequency that the user desire. Figure 6 illustrate the VI Front Panel.

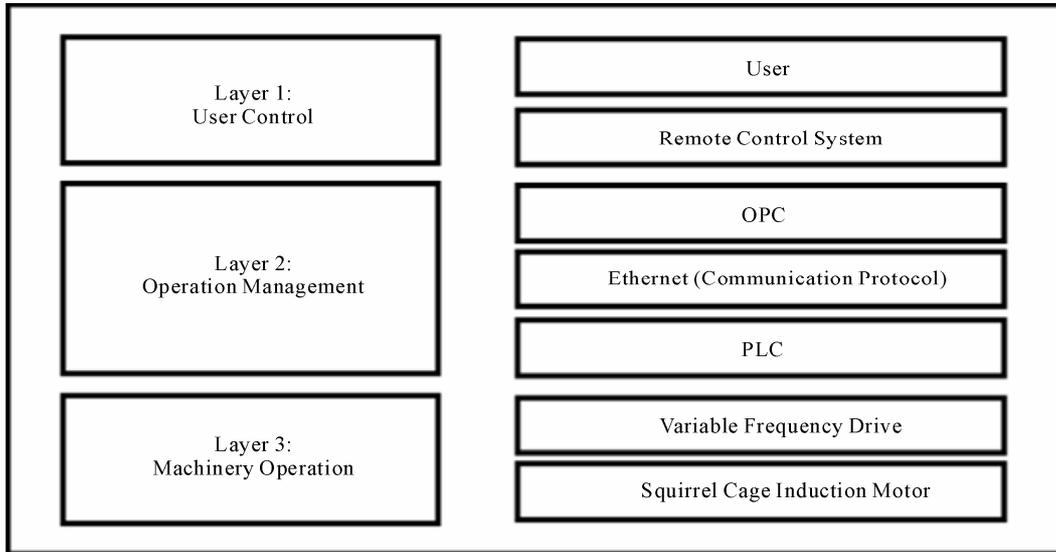


Figure 4. Three layer network architecture.

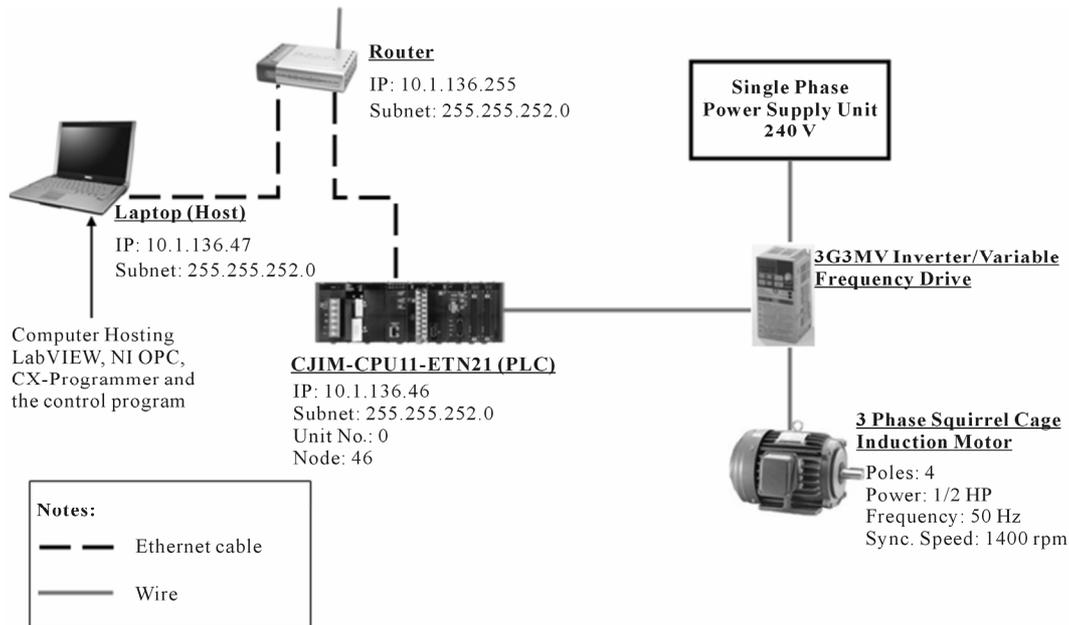


Figure 5. Actual network configuration.

Before running the VI, make sure all the hardwares have been switched on and configured correctly, and launch the NI OPC Quick Client so that the OPC tags can be browsed by the shared variables in this VI.

There are 5 OPC tags was created and used in the implementation, and the details of the tags have been tabulated in **Table 1**.

By referring to **Figure 6**, the Green push button is the switch to determine the motor is turning in forward direction. Meanwhile, the Orange push button is the switch to determine the motor is turning in reverse direction. Both Green and Orange light indicators at the right hand

side show whether the push buttons have been switched ON. The knob labelled as “Frequency” is the key program to control the frequency as well as the speed of the motor. The push button labelled as “Stop” function to stop the program execution.

Figure 7 illustrate the VI Block Diagram, which the programming part of the VI. By referring to the **Figure 7**, the Red colour square boxes with dotted lines are the VI components that are visible in both Front Panel and Block Diagram. For Example the push buttons, knob, light indicator. Meanwhile, the Red colour square boxes without dotted lines are the VI component that is visible

Table 1. The details of OPC tags and its connection to inverter.

OPC Tag name	PLC Address	3G3MV Inverter	
		Connected Terminal	Terminal Name
Outputbit 1	CIO0001.01	S1	Multi-function input 1 (Forward/Stop)
Outputbit 2	CIO0001.02	S2	Multi-function input 2 (Reverse/Stop)
Outputbit 5	CIO0001.05	S5	Multi-function input 5 (Multi-step speed reference 1)
Outputbit 6	CIO0001.06	S6	Multi-function input 6 (Multi-step speed reference 2)
Outputbit 7	CIO0001.07	S7	Multi-function input 7 (Multi-step speed reference 3)

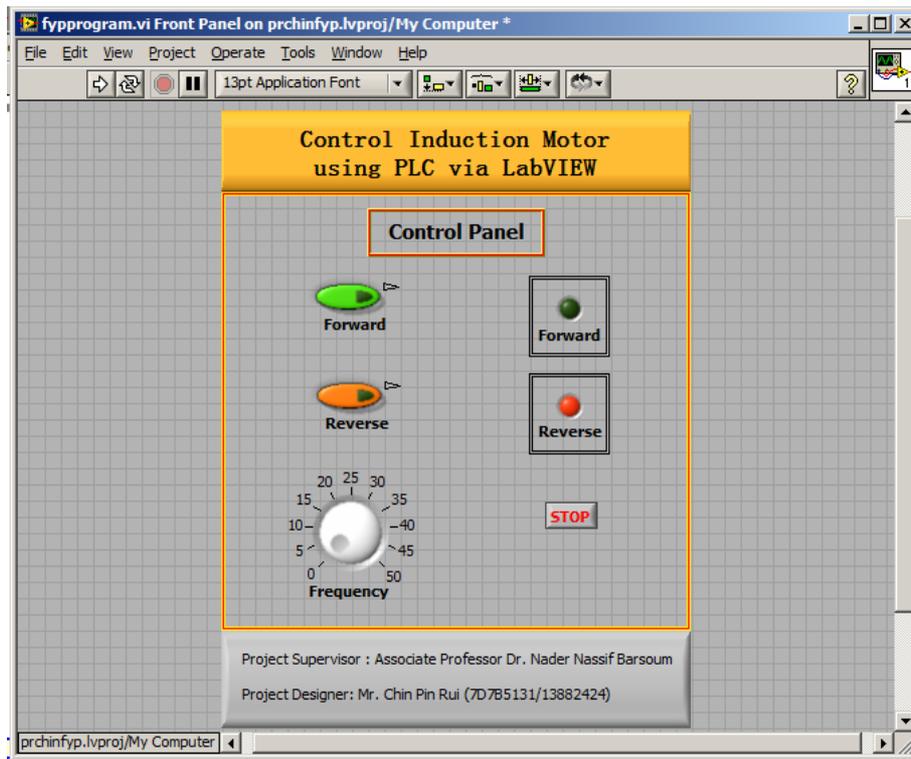


Figure 6. VI front panel of the project.

in Block Diagram but they are not visible in Front Panel, which are essential VI to structure the program. For example, shared variables, formula node, number to Boolean converter (consist of number to Boolean array and index array) and the While loop. The While loop, which is similar concept with the While loop in C Programming, is used in this VI to ensure the program execute continuously until the stop button is triggered.

The VI block diagram consists of two parts of the program. First part of the program is to allow user to switch on either Forward or Reverse direction of the squirrel cage induction motor. The second part of the program is to vary the frequency of the motor by changing the knob value.

By referring to **Figure 8**, if the user has switched on

the Forward button, the signal will be transmitted to the “outputbit1” shared variable, which is in write mode, *i.e.* the signal will be written into the PLC. Then once the signal has been successfully written to the PLC, the light indicator of the output address 1.01 of the PLC, and the “Forward” direction indicator in the LabVIEW’s Front Panel will light. This process identical to the reverse direction as well.

By referring to the **Figure 9**, the Red colour square boxes indicate the VI named formula node, where the language used in the formula node is C. After the user set the value of the frequency by turning the “Frequency Variable” knob, the value will be sent to the formula node, where it is denoted as “a”.

The output of the formula node has been divided into

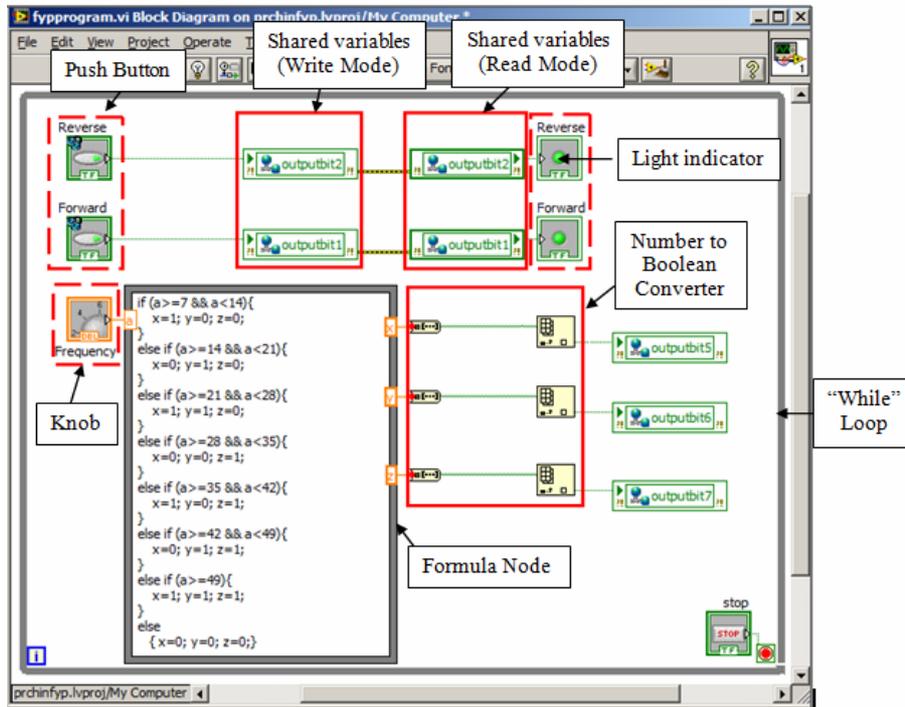


Figure 7. VI Block Diagram of the project.

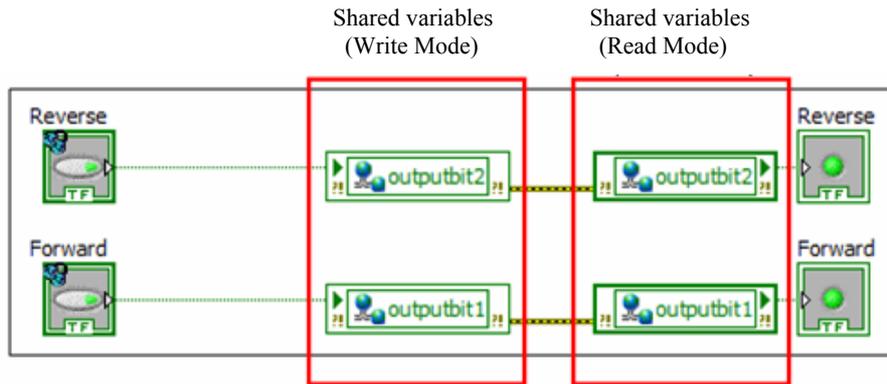


Figure 8. VI block diagram program (motor orientation switching part).

x, y and z, in which the number is either 0 or 1. In order to be readable by the shared variables, it is necessary to convert the number into Boolean format. When the signal is in Boolean format and sent to the shared variable, the information will be written into the PLC, and the light indicator of the output address 1.05, 1.06, 1.07 of the PLC will light up according to the desired output.

Table 2 below describes the relationship between the multi-step speed references 1 through 3 and frequency references 1 through 8.

The concept of frequency varying in Table 2 is crucial as the PLC communicates with the variable frequency drive in Boolean format. The value of each frequency refer-

ence can be set in the function parameters of the variable frequency drive. In this paper, the value of each frequency reference has been tabulated in Table 3.

By referring to Table 3, as long as the user has set the input frequency according to any of the condition, the number will be converting into Boolean in which it will arrange according to the relevant Multi-step speed reference. Then this Boolean number will trigger the variable frequency drive and determine the frequency and as well as the speed of the squirrel cage induction motor according to the output condition listed in Table 3. For example, if the user set the input as 8 Hz, in which the input range of this value is within $7 \leq x < 14$. Therefore, the Boolean number of Multi-step speed reference 1 is 1, and

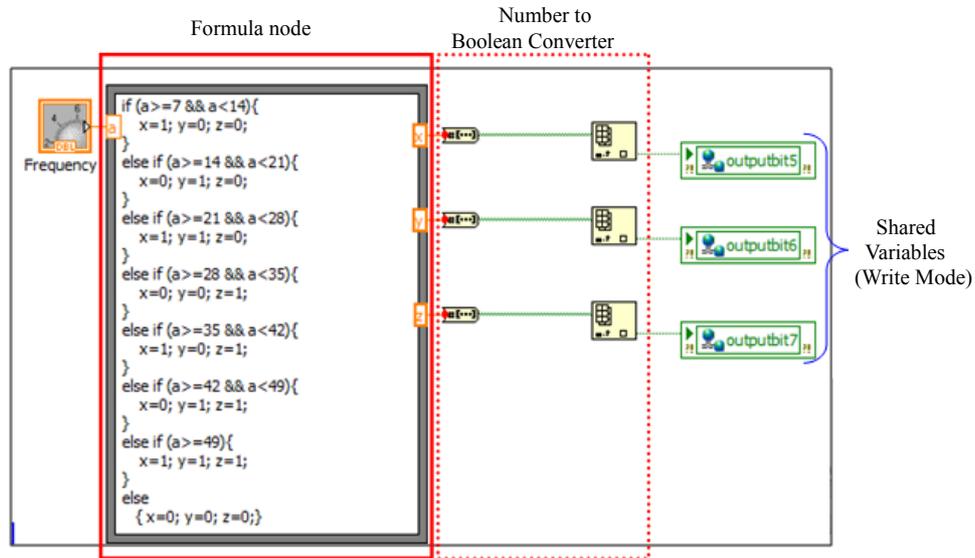


Figure 9. VI block diagram program (motor frequency varying part).

Table 2. Relationship between multi-step speed references and frequency reference [6].

Frequency reference	Multi-step speed reference 1 (Set value: 6)	Multi-step speed reference 2 (Set value: 7)	Multi-step speed reference 3 (Set value: 8)
Frequency ref. 1	OFF	OFF	OFF
Frequency ref. 2	ON	OFF	OFF
Frequency ref. 3	OFF	ON	OFF
Frequency ref. 4	ON	ON	OFF
Frequency ref. 5	OFF	OFF	ON
Frequency ref. 6	ON	OFF	ON
Frequency ref. 7	OFF	ON	ON
Frequency ref. 8	ON	ON	ON

Table 3. Conditions of input frequency in LabVIEW for varying frequency reference.

Frequency input range in LabVIEW (Hz)	Multi-step speed reference 1 (Outputbit 5)	Multi-step speed reference 2 (Outputbit 6)	Multi-step speed reference 3 (Outputbit 7)	Output frequency of squirrel cage induction motor (Hz)
$x < 7$	0	0	0	0
$7 \leq x < 14$	1	0	0	7
$14 \leq x < 21$	0	1	0	14
$21 \leq x < 28$	1	1	0	21
$28 \leq x < 35$	0	0	1	28
$35 \leq x < 42$	1	0	1	35
$42 \leq x < 48$	0	1	1	42
$x > 48$	1	1	1	49

the rest are 0. Then, this signal triggers the variable frequency drive to deliver the frequency of 7 Hz.

4. Testing and Verification

After the implementation, test is performed in order to verify that the system is working smoothly. **Figure 10** shows the front view of the system.

Figure 11 shows the PLC Ethernet Module light indicators that light up when the project VI is executing. And **Figure 12** shows the snapshot of the Squirrel Cage Three Phase Induction Motor when it is running.

5. Conclusions

In conclusion, the objective, scope and fundamental requirements of the project had been achieved. But there are some defects in this project. Firstly, there is a delay of time during the execution, which every decision made by the user through LabVIEW will take about 30 seconds in average to send the signal to the PLC. The delay mainly caused by the poor specification of the Laptop used for implementation, that leads the LabVIEW to take long time to execute the program and generate the Boolean data to send to the PLC.

Secondly, the Ethernet connectivity between the Laptop and the PLC is easily disconnected. This problem arises also due to the poor specification of the Laptop

used. This problem happen when the Laptop had used up too much RAM, causing the laptop lag and unable to use temporary. Therefore user cannot make too many actions and changes at the same time as it will crash the program. Thirdly, there is limitation in the speed control as there are only 3 multi-step references in the VFD. The frequency changes in this project cannot perform in smaller steps unless there are additional multi-step references provided.

In overall, choosing LabVIEW as the human machine interface for the implementation is a proper decision as it has various types of applications and functions that are easy to understand and use. Additionally, this approach is more economical as the objectives of the system implementation have been achieved with only basic functionality of the LabVIEW toolkits used, which are shared variables and NI OPC Servers.

OMRON CJ1 series PLC is easy to install and setup. Both hardware and software configuration can be easily done. It can carry out additional functions by simply add more units with various functions, like the Ethernet unit.

The 3G3MV inverter is a user friendly VFD that allows the user to configure the function parameter easily and the circuit wiring can be completed easily.

6. Acknowledgements

Special thanks to Daniel Wong and Jacintha Roland for

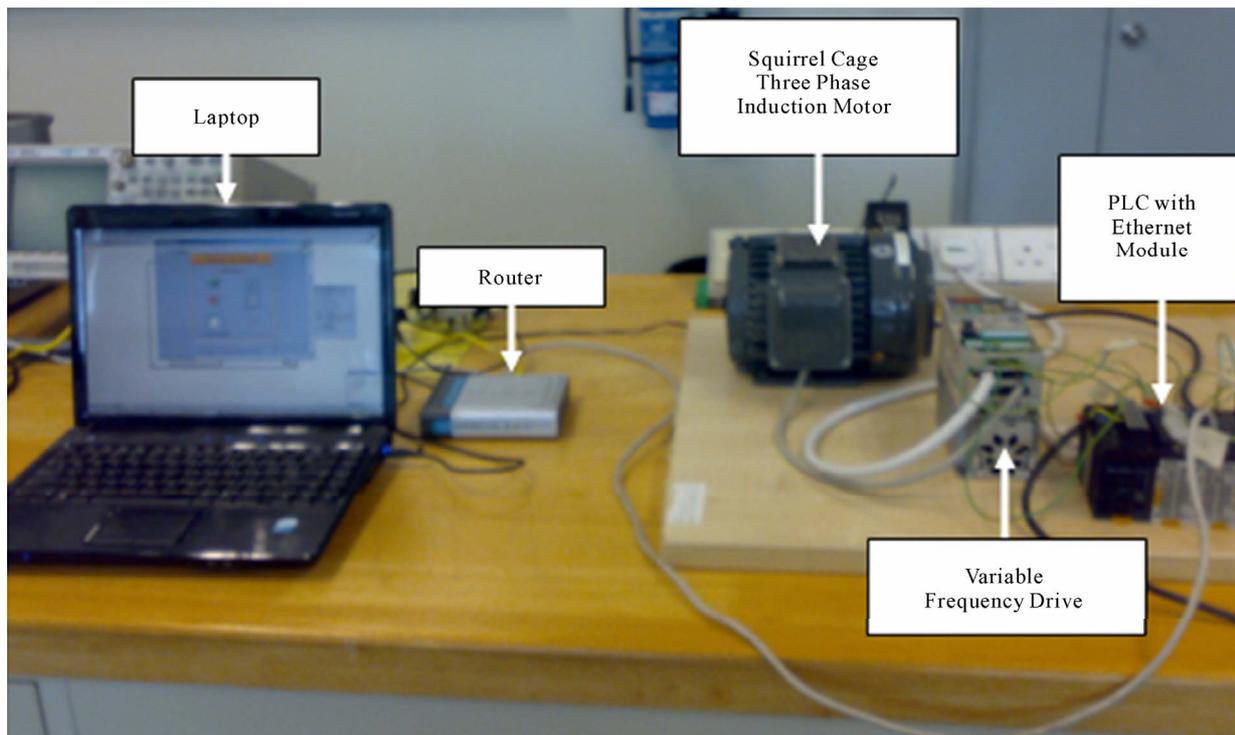


Figure 10. Front view of the project.



Figure 11. PLC ethernet module light indicator status.



Figure 12. Induction motor running in forward direction.

their remarkable advice throughout the design and implementation of the system.

7. References

- [1] W.-F. Chang, Y.-C. Wu and C.-W. Chiu, "Design and Implementation of a Web-Based Distance PLC Laboratory," *Proceedings of the 35th Southeastern Symposium on System Theory*, Morgantown, 16-18 March 2003, pp. 326-329. doi:10.1109/SSST.2003.1194584
- [2] M. Rodrigues, J. Mendes and J. Fonseca, "Application of a Web-Based Monitoring and Control System in Plastic Rotational Moulding Machine," *IEEE International Conference on Industrial Technology*, Hammamet, 8-10 December 2004, pp. 819-823
- [3] N. N. Barsoum and J. A. Roland, "Ethernet LabVIEW Control," Undergraduate Thesis, Curtin University Sarawak Campus, Sarawak, , 2010.
- [4] National Instruments, "Getting Started with LabVIEW," National Instruments, Technique Report 373427F-01, 2009.
- [5] National Instruments, "Using the LabVIEW Shared Variable," 2010. <http://zone.ni.com/devzone/cda/tut/p/id/4679>
- [6] N. N. Barsoum and Y. Y. Ng, "PLC Humidity Control Inverter Fed Induction Motor," Undergraduate Thesis, Curtin University Sarawak Campus, Sarawak, 2009.
- [7] OMRON Industrial Automation, "SYSMAC CJ Series Programmable Controller," OMRON, Technique Report W393-E1-14, 2009.
- [8] OMRON Industrial Automation, "CJ1M CPU Units with Ethernet Functions," OMRON, Technique Report, 2005.