Software Architecture of Ladder Compiller to Opcode for Micro PLC Based on ARM Cortex Processor

Muhammad A. Murti, Agung N. Jati, Lukam Mawardi, and Siti A.N.F. Agustina
Telkom University, Bandung, Indonesia
Email: {mam, ang}@ittelkom.ac.id, lukman.mawardi@gmail.com, aisha_florenza@yahoo.com

***Abstract*— Programmable Logic Controller (PLC) is an electronic circuit that can do a variety control function on complex levels and used as a substitute of mechanic relays components that used on control system. Over the past ten to fifteen years, many different programming languages have been used to program the PLC. For one programming language such as ladder diagram, each type of PLC have the rules and differences in the way of programming. In fact, modern industry usually uses not only one type of PLC alone but various types of PLC. It would be inefficient, both in terms of time and material. This paper discuss about ladder comppiler software design for micro class of PLC based on ARM processor. The architecture software was developed on the VB.Net platform. The designed software can work properly to compile ladder diagrams into commands/instructions and sent it to PLC. The design included status monitoring which is received I/O PLC status. The communications between ladder programmer software and PLC for each architecture part can work properly without any interruption with 100% reliable in terms of compiling from ladder diagram to ladder opcode. This software has been able to do the compiling process less than a second.**

***Index Terms*—Programmable Logic Controller, Ladder Programmer, Software Architecture**

I. Introduction

Programmable Logic Controller (PLC) is basically a controller specially designed to control a process or machine. Based on the number of inputs/outputs it has, generally PLC can be divided into 3 (three) groups [1]-[3]:

- PLC Micro (I/O PLC < 32 terminal);
- PLC Mini (I/O PLC ~ 32 -128 terminal);
- PLC Large /PLC Rack (I/O PLC > 128 terminal).

Generally, PLC consists of two main structuring components: Central Processing Unit (CPU) and interface system of input/output [1]-[4]. PLC works by reading the state of its inputs, for later used to change the state of its output. The changes that occurred in the PLC output is depending on the program. There is a compiler that compiles the ladder diagram to be opcode that will be loaded on PLC memory. PLC's processor needs Operating System (OS) to read and interpret opcode and then to execute program. During in its process, PLC conducts three main operation [1]-[3]:

- Reading input data from external equipment via input module;
- Executing a logic control program stored in the memory of PLC;
- Updating the data in the output module.

Besides, PLC also consists of software element which will construct ladder diagram or mnemonic code. That code will be sent to PLC using programming device to trigger instruction in PLC such as read data input or write data to output/memory.

Ladder diagram is the easiest program leanguage to user for develop controlling rule on PLC. Ladder diagram describes the instructions like Boolean/Logic operations and switching operations. There are some samples of ladder diagram such as normally open/close, add, compare, coil, and, or, etc [5].

This paper discuss as follows: second section will discuss about the architecture of ladder diagram programmer software; in the third section will be continued by discussing the result of implementation and experiment; and the last section will contain conclusions and suggestions.

II. System Modelling

A. Ladder Programmer

Fig. 1 show the shoftware architecture of Ladder programmer with four main menus.

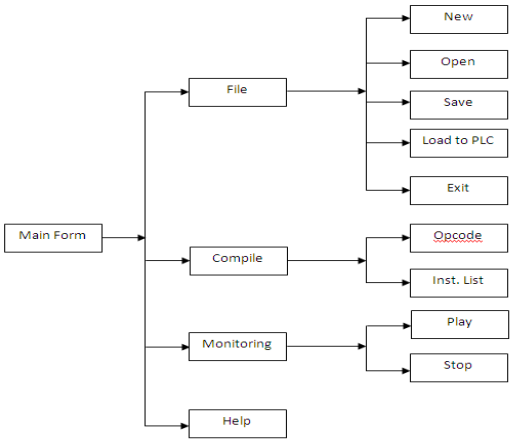

Figure 1. Software Architecture

Manuscript received October 15, 2013; revised December 13, 2013.

doi: 10.12720/joace.2.4.363-366

This Ladder Programmer Software has several functions as follows :

- As a GUI between PLC and user, it will show ladder design and I/O status of PLC
- Compile the ladder diagram to ladder opcode and instruction list
- Simulate processes
- Control the PLC, and consists instructions such as PLAY, STOP, MONITOR, LOAD TO PLC, and PLC PROGRAMMING
- Handle data communication using RS232 format.

This software was developing on VB.Net platform because more compatible with many Windows OS, support GDI+ to build graphical pictures and text, and also reduce the VGA load.

To develop the software, there was several steps in developing the compiller. They are:

- Build serial communication based on UART Protocol
- Build the interface of Main Form
- Design the algorithm of Ladder Diagram Compilation
- Design of Compiling Procedures
- Build the Output Interface
- Build the Monitoring Status Interface

B. *Ladder Comppiller Concept*

The ladder compiller program software has the following working flow as can be seen on Fig. 2.

Fig. 2 shows all procedure which designed on VB.net. One of all is the serial communication procedure. This procedure contains the UART protocol, so the software can transfer data from / to PLC. We design the sofware work with baudrate 19200 bps.

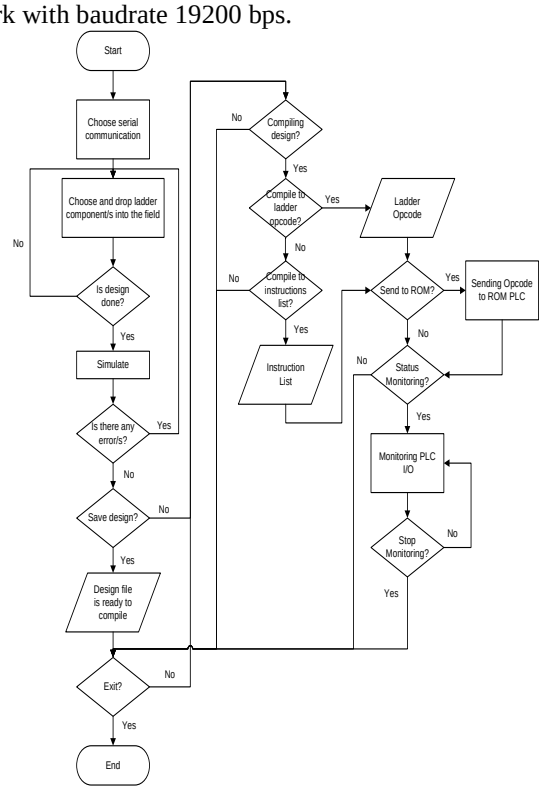

Figure 2. Software design flow diagram

Fig. 2 shows all procedure which designed on VB.net. One of all is the serial communication procedure. This procedure contains the UART protocol, so the software can transfer data from / to PLC. We design the sofware work with baudrate 19200 bps.

Other procedure is designed as like as usual given from the platform to simplify the software architecture. This research is focused on ladder to opcode comppiler design.

C. *Ladder to Opcode Comppiler Design*

In this procedure, we described encode-decode process from ladder diagram or instruction to binary opcode. Opcode construct by three binary code ### and followed by <data>. Three binary code provide possiblility to compille up to 1000 instructions. Table I contains common use instructions list that developed on this work.

Design page only use one picturebox and every cell create by lines from GDI+ makes every cell not deffine by array picturebox, but array region. Region is a velue thet represent viewer coordinate area from point (x1,y1) to (x2,y1), and (x1,y2) to (x2,y2)). The program will find region that contain ladder component and compille the ladder to appreciate opcode.

TABLE I. OPCODE FORMAT

Instructions	*Opcode*	*Ladder Opcode Format*
LOAD	000	000<data>
LOAD NOT	001	001<data>
OR	002	002<data>
OR NOT	003	003<data>
AND	004	004<data>
AND NOT	005	005<data>
OR *LOAD*	006	006<data>
AND *LOAD*	007	007<data>
OUTPUT	008	008<data>
OUT NOT	009	009<data>
TIMER	020	020<timer value>
COUNTER	022	022<counter value>
COMPARE	030	030<data><data2>
ADD	070	070<data><data2>
SUBSTRACT	071	071<data><data2>

Ladder opcode operand word will be classified into three categories below:

- If the first digit is “0”, then the operand is in BCD format
- If the first digit is “1”, then the operand is in Binary format
- If the first digit is “2”, then the operand is from register.

Table II contain the data memory area as designed PLC Format.

TABLE II. DATA AREA PLC FORMAT

Instructions	Input Area	Output Area	Timer/Counter Area
LOAD, *LOAD* NOT, OR, OR NOT, AND, AND NOT, OR *LOAD*, AND *LOAD*	000000-000715 (bit)	000800-001215 (bit)	
OUTPUT, OUTNOT		000800-001215 (bit)	
TIMER (TIM)			0097 – 0160 (word)
COUNTER (CNT)			0233 – 0296 (word)
COMPARE (CMP)	0000-0007 (word)	0008-0012 (word)	0097-0224/0233-0360 (word)
ADD (ADD)	0000-0007 (word)	0008-0012 (word)	0097-0224/0233-0360 (word)
SUBSTRACT (SUBS)	0000-0007 (word)	0008-0012 (word)	0097-0224/0233-0360 (word)

III. RESULTS AND DISCUSSION

A. Opcode Decoding

There are some tests to prove that all of the opcode instructions are true, compare with the design. All of the tests only use the simplest operation, which is consist two operands for each ladder. The reslut of tests can be seen on Table III.

TABLE III. .LADDER TO OPCODE

No	Ladder	Opcode
1	OR	0000000000040000010080008000@
2	AND	0000000000030000010090008000@
3	Timer	0000000000212000100000000100800080 0@
4	Counter	0000000000000000010222000000000000200800 0800@
5	ADD	0000000030702000020001200 02@
6	SUBSTRACT	0000000050712000520006200 07@
7	COMPARE	0000000020302000820009000004705008000800 @

From that test, It's proved that the opcode are matched with the initialization design. The size of the opcode depends on the operand number and each value. For every opcode is always ended by "@" as designed.

B. Time Process

Time process is very important parameters to prove that the system is reliable. Time process also can show how the system works to compile ladder into opcode. We can say that the faster compilation is better for user.

We measure the time process by insert timer on program. Timer wil begin by single click on compile order and stop on opcode has resulted. Table IV show the averrage time process for several instructions.

TABLE IV. TIME PROCESS COMPILATION

No	Instructions	Average Time Process (ms)
1	OR	1.566
2	AND	1.636
3	Timer	0.0818
4	Counter	0.1284
5	ADD	0.0384
6	SUBSTRACT	0.0429
7	COMPARE	0.0884

Logical operation need more time to compile than other operation. It's because in the logical operation system must convert the hexadecimal operand into binary operand first and the execute them. Time proccess for single instruction has less then 2 ms.

C. Load Process

Load process test is to prove that the opcode as compiled result can be send to PLC ROM and readable by PLC processsor properly. It also proves that serial communication is work properly.

The scenario is trying to send simplest opcode. There is the ladder diagram which only consist an input and an ouput. See the picture below.

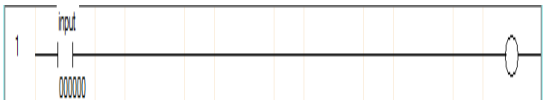

Figure 3. Simplest ladder diagram

To send the opcode, first software will send a command #LOADCPURDY and then the opcode. If load opcode to PLC success, PLC will respond by replying software #LOADCPUSUCCESS and if not then PLC will reply #LOADCPUOVERLOAD.

Opcode result from the ladder diagram above is 000000000008000800@. It will be sent into PLC and show that the load success without any interruption. Time needed to send complite opcode to PLC is 19.734 ms.

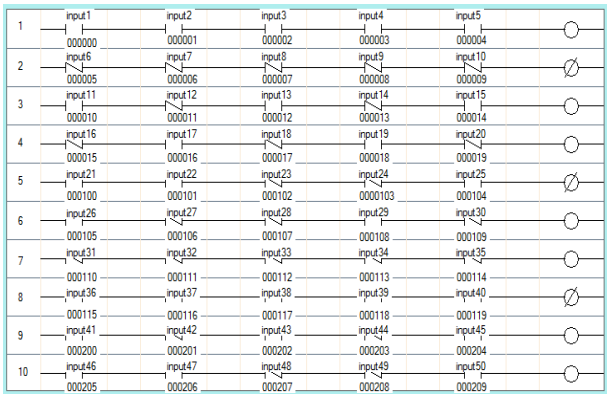

Figure 4. Longer ladder diagram load test

Next test was with long ladder diagram showed in the Fig. 4. The ladder diagram convert into opcode 000200000022000100320002003200030022000400821 0

000012000500320006002200070032000800220009009210010002000900320010002200110032001200921002002200130002001400220015003200160082100300320017002200180012001900320020008210040032002100320022003200230002002400821005002200250022002600220027002200280082100500020029003200300022003100320032002200330092100600120034002200350032003600220037009210070032003800020039002200400032004100821008003200420032004300120044003200450082100900220046002200470032004 8@.

The result showed that the opcode was success to send into PLC without any interruption. To send this opcode, it needed longer transmition time than simplest ladder opcode. Trinsmistion time for long opcode was 453 ms.

III. Conclusions

Design of ladder diagram software compiler for PLC based ARM architecture CPU using VB.Net platform can work properly without any interruption. The time process is also reliable. And the most important thing which is the compiling process has 100% matched without errors. The time process to compile and to load is still reliable because it's less than 1 s.

The next project from this research is to make user interface more user friendly. So it will be more interesting and easier to control and monitor the PLC based on ARM processor.

Acknowledgment

This work was supported in part by a grant from Indonesia Higher Education Department with RAPID 2013 funding program.

References

[1] I. Setiawan, *Programmable Logic Controller (PLC) dan Teknik Perancangan Sistem Kontrol*, Andi Publisher, Semarang, 2005.
[2] L. Mawardi and M. A. Murti, "Perancangan dan implementasi PLC mikro berbasis mikrokontroler ST uPSD3254BV," Thesis of Institut Teknologi Telkom, Bandung, 2009.
[3] M. A. Murti, S. Sumaryo, and L. Mawardi, "Operating design for micro PLC based on ARM cortex 3," in *Proc. International Conference on Computing and Information Technology*, 2012.
[4] R. Ramadhani, "Perancangan dan realisasi sistem minimum programmable logic controller (PLC) berbasis mikrokontroler atmel AT89S8252," Thesis of Sekolah Tinggi Teknologi Telkom, Bandung, 2005.
[5] H. Mukharrydani, "Perancangan dan realisasi sistem otomasi akuarium berbasis mikrokontroler AT89C52," Thesis of Sekolah Tinggi Teknologi Telkom, Bandung, 2007.

Muhammad A. Murti was born in Jakarta on June 7th, 1975. He Graduated from Universitas Brawijaya at 1998 on Electrical Engineering and Master of Engineering on Instrumentation and Control from Institut Teknologi Bandung at 2004.
He joined STTTelkom (currently Telkom University) in 1998 as Lecturer and since it he has worked in various functions at STTelkom/ITTelkom/Tel-U including Coordinator at Electrical Measurement Lab, head of Electronic System Laboratory, Control and Industrial Automation Laboratory, and head of Undergraduate Program (2007-2011).
He involved at many conference, such as WOCN2007 Singapore, WOCN2008 Surabaya, ICCSII2012, TIME-e2013, APCC2013 and more. He is a member of IEEE since 2005, IEEE Communication Society, IEEE Control System Society and IEEE Robotics and Automation Society. He interest on Electrical, Control, and Industrial Automation. Currently he is working on Networked Control System Modeling and Cellular M2M.

Agung N. Jati was born in Kendal on January 16th, 1988. He joined to IT Telkom (now called Telkom University) in 2010 as a lecturer for Telecommunication Engineering until he was transfered to Computer Engineering. He also has been placed as a coordinator of Robotics Research Group and Embedded System Laboratory since 2011.
He graduated from Institut Teknologi Telkom at 2009 on Bachelor Degree of Telecommunication Engineering and achieved a Master of Electrical Engineering from the same institution at 2012.
Since he was a lecturer, he ever researched about embedded system twice(funded by IT Telkom), and PLCs (funded by Ministry of Education and Art).

Lukman Mawardi was born in Bandung, West Java, Indonesia on May, 20th 1978. He attended his basic education from elementary until high school in his hometown, Bandung. He attended college for Bachelor's Degree in Telkom Institute of Technology in Bandung, West Java, Indonesia, majored in Telecommunication Engineering. He graduated at 2009.
He ever did a practical job in Research and Development Center of Mineral and Coal Techology, Ministry of Energy and Mineral Resources (Puslitbang tekMIRA ESDM) for about 2 years and PT Telkom Indonesia for about 1 year. At Puslitbang tekMIRA ESDM, he developed an embedded system of kaolin, feldspar, and gold mining automation and monitoring, using microcontroller system, dataLogger device, PLC, and desktop based software. In PT Telkom Indonesia, he did a research of modification, development, and "performance rising" of public payphone firmware, which installed and used in West Java Region. Now he is currently working in PT Elda Sarana Informatika, an IT and embedded system development company, located in Bandung, West Java, Indonesia. He has been working at that place since 2007. Now he works as Software Division Manager.

Siti A.N.F. Agustina was born in Bogor, Indonesia on December, 5th 1990. She graduated at 2012 with exceed expectation from Telkom Institute of Technology in Electrical Engineering.
During her college, she did a practical job at Diskominfo Bandung and PT Pindad as a web designer, assistant of Engineering staff in Industrial Machinery and Services Division. She was also a programmer for PLC using ladder diagram and Visual Basic. Currently she is working in PT Telkom Indonesia as Access Unit staff. She supports the Management Data in terms of creating a network on a web called TeNOSS.
She was a member of IEEE Student Branch Telkom Institute of Technology, Bandung, Indonesia.