



**Birzeit University**  
**Faculty of Engineering & Technology**  
**Department of Electrical & Computer Engineering**  
**ENEE5102-Power Lab**

**“Introduction to Siemens PLCs”**

**Student: Mohamad Bornat**

**Instructor:Dr.Ahmad Alyan**

**April, 2018.**

# *Abstract*

This aim of this experiment is to get familiarized with Siemens Programmable Logic Controllers (PLC) as hardware and the software used to program. Different PLCs operations such as PLC timed operation, Counter operation are introduced. The PLC program can be written in many languages for example, Ladder diagram, Statement List, Sequential Function Charts, and Function Block Diagram. In this experiment Ladder diagram and Statement List are used. The PLCs are widely used in the field of protection and automation in electrical systems.

# Contents

<b>Acronyms and Abbreviations</b> .....	<b>iv</b>
<b>List of Figures</b> .....	<b>v</b>
<b>Chapter 1 Introduction</b> .....	<b>1</b>
<b>Chapter 2 Procedure and Discussion</b> .....	<b>3</b>
2.1 Preparing the Software .....	3
2.2 Set and Rest Coil.....	4
2.3 PLC Timers .....	5
2.4 PLC Counters.....	7
2.5 Comparison Instructions .....	9
<b>Conclusion</b> .....	<b>11</b>
<b>References</b> .....	<b>12</b>
<b>Appendices</b> .....	<b>13</b>

# Acronyms and Abbreviations

PLC	Programmable Logic Controller
LAD	Ladder Diagram

# List of Figures

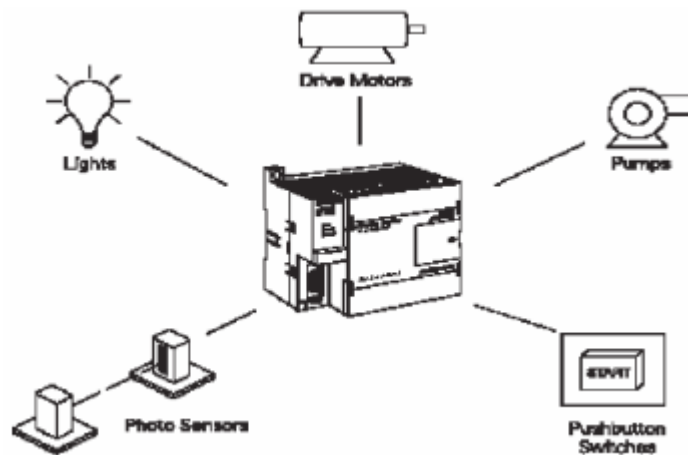
<i>Fig. 1.1:</i> PLC Operation Environment .....	1
<i>Fig. 2.1:</i> LAD For Set and Reset.....	4
<i>Fig. 2.2:</i> Ladder Diagram of PLC Timer. ....	6
<i>Fig. 2.3:</i> Ladder Diagram of PLC Counter .....	8
<i>Fig. 2.4:</i> CUD with GEQ instruction .....	9

# Chapter 1

## Introduction

---

Many manufacturing operations are ON/OFF in nature, that is a conveyor or heaters is either on or off, a valve is either open or closed, and so on. In the past, these types of discrete control functions were often provided by a system of electrical relays wired according to a complex diagram into what was called a relay logic system. In recent years, computers have also taken over the operation of such relay logic controllers, known as programmable logic controllers (PLCs). Even though originally designed to control discrete state (ON/OFF) systems, they are capable also to control analog control loops. A PLC monitors inputs, makes decisions based on its program, and controls outputs to automate a process or machine as shown in figure 1.1 .This section is meant to supply the reader with basic information on the functions and configurations of PLCs.



*Fig. 1.1:* PLC Operation Environment

Typically a PLC system has five basic elements:

- Input modules: 5 V, 24 V, 110 V, 240 V DC
- Output modules: relay type (ac and dc switching), transistor type (dc switching), or triac type (loads with ac power supply)

- Central processing module (CPU) + memory
- Power supply module
- Programming device
- Operator interface (optional)

The *input module* accepts a variety of digital or analog signals from various field devices (sensors) and converts them into a logic signal that can be used by the CPU. The *CPU module* makes decisions and executes control instructions based on program instructions in memory. *Output modules* convert control instructions from the CPU into a digital or analog signal that can be used to control various field devices (actuators). A *power supply module* is needed to convert the main a.c. voltage to necessary voltage level for the processor and the circuits in the input and output modules. A *programming device* is used to input the desired instructions in the memory of the processor. These instructions determine what the PLC will do for a specific input. An *operator interface* device allows process information to be displayed and new control parameters to be entered. This module is optional and may be not found in some applications.

The corresponding software to the used PLC must be installed on the programming device like as PC. Then the hardware of the PLC must be defined for the software, this step is called hardware configuration. Control programs can be implemented in a Project of the software and then downloaded to the already configured CPU of the PLC. After providing the CPU with the program, the PLC will be ready to control a process[2].

# Chapter 2

## Procedure and Discussion

---

### 2.1 Preparing the Software.

Firstly , the project was created by selecting the menu item File→New , a hardware item was assigned to the project with the menu item Insert→Station→2 SIMATIC 300 Station , after that the hardware was defined by going to the the Hardware Configurator call up the hardware catalog with Insert □ Hardware Components. In this box, select the C7-621from SIMATIC 300 □ C7.

Ladder Logic was used first to program the PLC, Ladder logic uses components that resemble elements used in a line diagram format to describe hardwired control. The left vertical line of a ladder logic diagram represents the power or energized conductor. The output element or instruction represents the neutral or return path of the circuit. The right vertical line, which represents the return path on a hard-wired control line diagram, is omitted. Ladder logic diagrams are read from left to-right, top-to-bottom. Rungs are sometimes referred to as networks. A network may have several control elements, but only one output coil.

How to program in LAD

1. Under View, select symbolic representation.
2. At OB1, enter your program. At Network 1, enter connecting in series. For connecting in parallel left click on the left Line of the Network 1 and then right click to select Branch Open. After connecting the desired elements in the second Branch you have to connect the right side of this Branch with the suitable Point in the first Branch.
3. For inserting program elements there are three methods: a. Click on the desired icon on the toolbar b. Right-click on the circuit to open the pop-up menu. Select what do you need. c. Drag the coil to your circuit using the drag-and-drop operation.



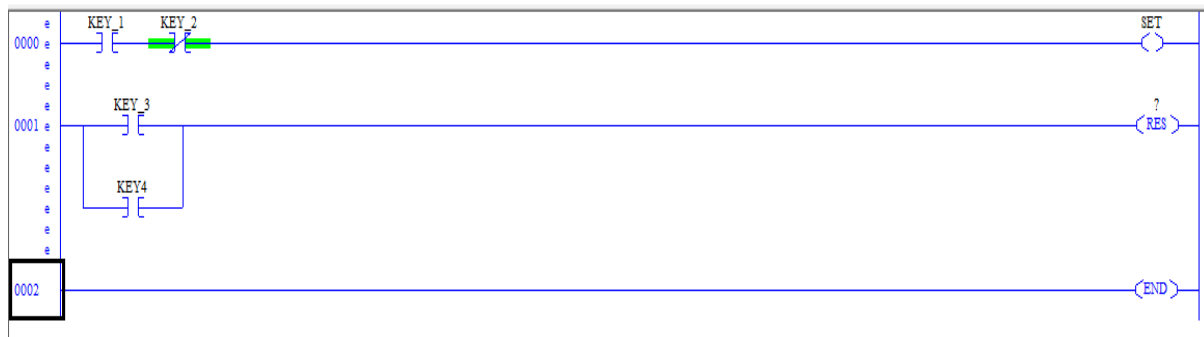
4. you have to name and address the contacts and coils. Click on the ???. Enter the symbolic name “any name” (with quotation marks). Then click outside the circuit to get the edit symbol, where you can address the symbolic name.

A statement list (STL) was used as well, which provides another view of a set of instructions. The operation, what is to be done, is shown on the left. The operand, the item to be operated on by the operation, is shown on the right. A comparison between the statement list shown below, and the ladder logic shown on the previous page, reveals a similar structure. The set of instructions in this statement list perform the same task as the ladder diagram.

## 2.2 Set and Rest Coil

A function that is different of output function often available is the ability to set and reset an internal relay. These instruction causes the relay to self-hold, that is, latch. It then remains in that condition until the reset instruction is received. The term flip-flop is often used. The SET coil is switched on when power is supplied to it and remains set until it is RESET. The RESET coil is reset to the off state when power is supplied to it and remains off until it is SET.

The following LAD was written and executed on the PLC S7 :



*Fig. 2.1: LAD For Set and Reset.*

One normally closed and one normally open contacts closures were placed in the first instruction combination and two normally open contact closures were placed in parallel in the second instruction Combination. Key 1 , Key 2 and Set Coils represent the first instruction combination. If input Key1 is energized, output set coil is latched. The inputs could be switches, pushbuttons, or contact closures. Key 3, Key4 and Reset Coil represent the second instruction combination. If either input is energized, output reset coil will unlatch the output..

- ❖ The same program was executed on SIMATIC S5-100U PLC using statement list programming as in code (A2.1). Then the operation was tested and it gave the same results.

## 2.3 PLC Timers

Timers are represented by boxes in ladder logic. When a timer receives logic “1” at the start input (S), the timer starts to time. The timer compares its current time with the pre-set time value (TV). The output (Q) of the timer is logic 0 as long as the current time is less than the pre-set time. When the current time is greater than the pre-set time the timer output is logic 1.

On-Delay timer starts the specified timer if there is a positive edge at the start input (S). A signal change is always necessary in order to enable a timer. The timer runs for the time interval specified at input (TV) as long as the signal state at input (S) is positive. The signal state at output (Q) is “1” when the timer has elapsed without error and the signal state at the (S) input is still “1”. When the signal state at input (S) changes from “1” to “0” while the timer is running, the timer is stopped. In this case the signal state of output (Q) is “0”.

The timer is reset if the reset input (R) changes from “0” to “1” while the timer is running. The current time and the time base are set to zero. The signal state at output (Q) is then “0”. The timer is also reset if there is a logic “1” at the (R) input while the timer is not running and the RLO (Result of logic operation) at input (S) is “1”. The current time value can be scanned at the outputs (BI) and (BCD). The time value at (BI) is binary coded, at (BCD) is BCD coded. The current time value is initial (TV) value minus the time elapsed since the timer was started.

The Off-Delay Timer (TOF) is used to delay turning an output OFF for a fixed period of time after the input turns OFF. When the enabling input turns ON, the timer bit turns ON immediately, and the current value is set to 0. When the input turns OFF, the timer counts until the elapsed time reaches the preset time. When the preset is reached, the timer bit turns OFF and the current value stops counting. If the input is OFF for a time shorter than the present value, the timer bit remains ON. The TOF instruction must see an ON to OFF transition to begin counting.

The following ladder diagram was written, and then its operation was tested using PLC SIMATIC C7-621.:

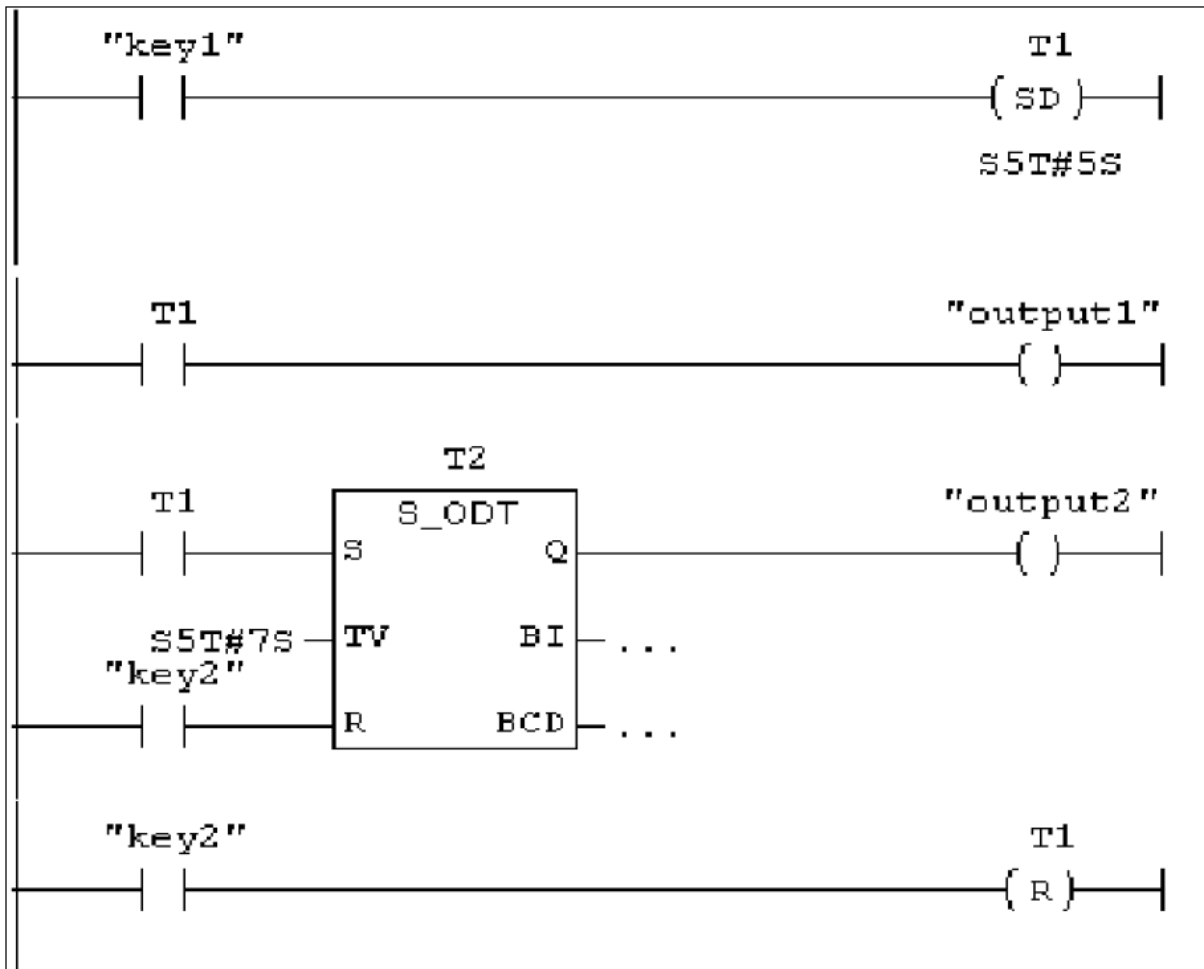


Fig. 2.2: Ladder Diagram of PLC Timer.

Consider a switch is connected to input Key1 and a light is connected to output2, when Key one is pressed (changing its signal state from “0” to “1”) the timer T1 will be started, output1 is latched. The pre-set time value (TV) has been set to 5. The associated contact T1 will be closed after 5 seconds and the light will turn on as long as the switch key1 is still closed. If the switch Key 1 was opened before 15 seconds had passed, the timer is stopped and the light is turned off .(if the signal state of the reset input Key2 changes from “0” to “1”, the time is reset regardless of whether the timer is running or not).

- ❖ The same program was created on SIMATIC S5-100U PLC using statement list programming as in code (A2.2). Its operation was tested and gave the same results of LAD.

## 2.4 PLC Counters

Counters used in PLCs serve the same function as mechanical counters. Counters compare an accumulated value to a pre-set value to control circuit functions. Control applications that commonly use counters include the following:

- Count to a pre-set value and cause an event to occur
- Cause an event to occur until the count reaches a pre-set value A bottling machine, for example, may use a counter to count bottles into groups of six for packaging.

Counters are represented by boxes in ladder logic. Counters increment/decrement one count each time the input transitions from off (logic 0) to on (logic 1). The counters are reset when a RESET instruction is executed. There are three types of counters: up counter (CU), down counter (CD), and up/down counter (CUD).

The Count Up (CTU) instruction counts up from the current value each time the count-up input CU makes the transition from off to on. When the current value (Cxxx) is greater than or equal to the Preset Value (PV), the counter bit (Cxxx) turns on. The counter is reset when the Reset (R) input turns on, or when the Reset instruction is executed. The counter stops counting when it reaches the maximum value (32,767). Its ranges: Cxxx=C0 through C255

The Count Down (CTD) instruction counts down from the current value of that counter each time the count down input CD makes the transition from off to on. When the current value Cxxx is equal to zero, the counter bit (Cxxx) turns on. The counter resets the counter bit (Cxxx) and loads the current value with the preset value (PV) when the load input (LD) turns on. The Down Counter stops counting when it reaches zero, and the counter bit Cxxx turns on. Its ranges: Cxxx=C0 through C255.

The following ladder diagram was created, and then its operation was tested using PLC SIMATIC C7-621:

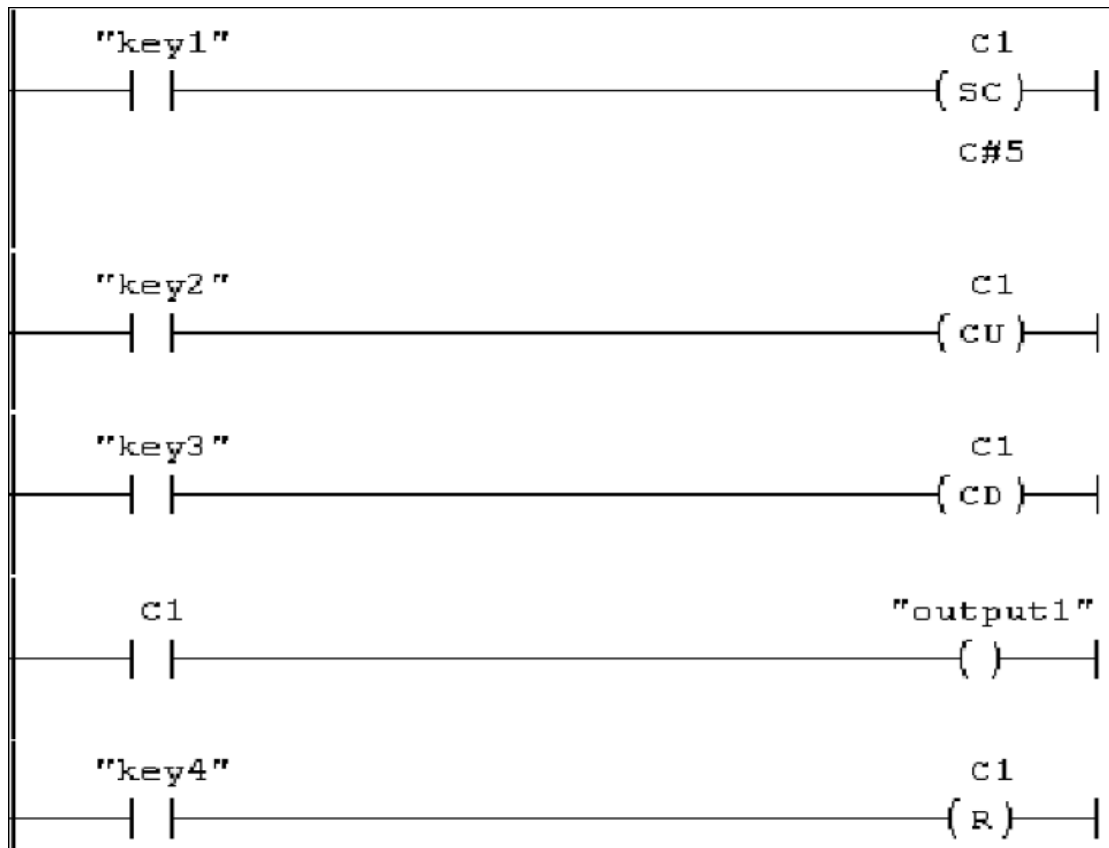


Fig. 2.3: Ladder Diagram of PLC Counter

If Key1 is pressed the set input (S) changes from logic 0 to logic 1, the counter is preset with the value of 5. If the signal state of (CU) changes from logic 0 to logic 1, the value of counter (C no.) will be incremented by one – except when the value of (C no.) equal to “999”. If (CD) changes from logic 0 to logic 1, the value of counter (C no.) will be decremented by one - except when the value of (C no.) equal to “0”. The output is “1” if the current value (C no.) is not equal to zero. When Key4 is pressed the counter resets its values.

- ❖ The same program was created on SIMATIC S5-100U PLC using statement list programming as in code (A.3). Its operation was tested and gave the same results of LAD.

## 2.5 Comparison Instructions

PLC's have special input instructions that are used to perform comparison functions. For example, the PLC on your trainer includes the following comparison instructions:

- Equal (EQU);
- Not Equal (NEQ);
- Less Than (LES);
- Less Than or Equal (LEQ);
- Greater Than (GRT);
- Greater Than or Equal (GEQ).

Each of these instructions compares a pair of arithmetical values. The result of the comparison determines whether the comparison instruction is true or false.

The following ladder diagram was written, and then its operation was tested using PLC SIMATIC C7-621.:

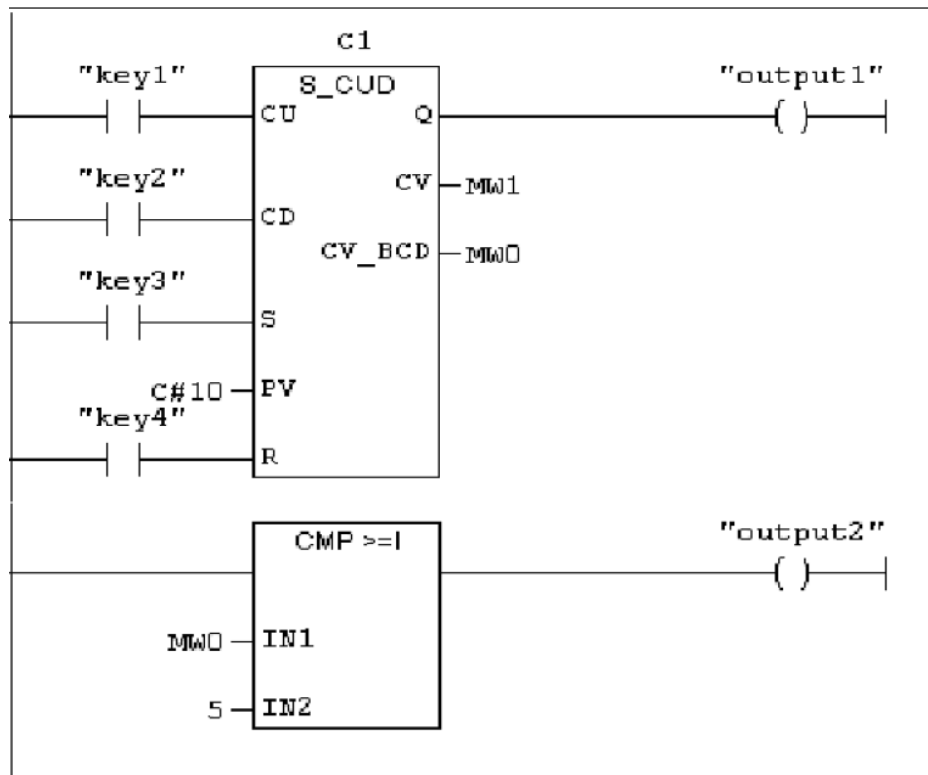


Fig. 2.4: CUD with GEQ instruction

The first instruction of this program is an Up/down counter, when the signal state at the set input (S) changes from logic 0 to logic 1. The counter output has been directed to output1. As Key1 is pressed, the count is incremented by one. As Key2 is pressed. The count is decremented the count by 1. When the count has reached zero, the output 1 turns off. And in the second branch a comparison instruction is placed to energize output 2 if necessary, this instruction determines whether the value at Source A is greater than or equal 5. If so, the GEQ instruction is true and output 2 is energized. If not, the GEQ instruction is false. Key 4 is a reset key.

- ❖ The same program was created on SIMATIC S5-100U PLC using statement list programming as in code (A.4). Its operation was tested and gave the same results of LAD.

# Conclusion

---

To sum up, this experiment gave an introduction to Siemens PLCs, which are used in many fields in practice, such as protection and automation in electrical systems, industrial life and controlling systems. Several operations were introduced such as internal relay, PLC timed operation, Counter operation, comparison instruction. These operations were implemented using Ladder diagram and statement list languages, and it was tested on SIMATIC S5-100U PLC and PLC SIMATIC C7-621.



# References

---

- [1] Beards C.F., Vibrations and Control System. Ellis Horwood, 1988.
- [2] Bennett S., A History of Control Engineering. 1800-1930. IET. 142–148, June 1986.
- [3] L. A. Bryan, E. A. Bryan, “Programmable Controllers Theory and Implementation “ 2<sup>nd</sup> ed. 1997.

# Appendices

---

## A2.1

AI0.0  
ANI0.1  
SQ1.0  
OI0.2  
OI0.3  
RQ1.0

## A2.2

AI0.0  
LKT5.2  
SRT1  
AT1  
=Q1.0  
AT1  
LKT7.2  
SRT2  
AI0.1  
RT2  
  
AT2  
  
=Q1.1  
AI0.1  
RT1

### A2.3

13  
A I 0.0  
L K C 5  
S C 1  
A I 0.1  
C U C 1  
A I 0.2  
C D C 1  
A C 1  
= Q 1.0  
A I 0.3  
R C 1

### A2.4

A I 0.0  
C U C 1  
A I 0.1  
  
C D C 1  
A I 0.2  
L K C 10  
S C 1  
A I 0.3  
R C 1  
A C 1  
= Q 1.0  
L C 1  
L K C 5  
>=F  
= Q 1.1