Lab
C1118

كلية الهندسة – جامعة المنصورة

# C1118 معمل التحكم التعاقبي

Computer Engineering and Control Systems

قسم هندسة الحاسبات ونظم التحكم

# Laboratory Book

COMPUTER ENGINEERING AND CONTROL SYSTEMS DEPARTMENT

C1118 معمل التحكم التعاقبي

# Laboratory Book

# Table of Contents

Part 1

# 1: Laboratory Basic Information

## أولا : البيانات الأساسية للمعمل

| | |
|---|---|
| إسم المعمل: | التحكم التعاقبى (الرقم الكودي : C1118 ) |
| القسم العلمي: | هندسه الحاسبات والنظم |
| المشرف: | د/مصطفى الحسينى |
| مهندس المعمل: | لا يوجد |
| أمين المعمل: | السيده/ فاتن عبد الغفار محمد |
| التليفون: | داخلي 1305 |
| الموقع بالنسبة للكلية: | الناحية البحرية |
| مساحة المعمل: | 120 متر |

Part 2

## 2: Laboratory Instruments

## ثانياً: قائمة بالأجهزة والمعدات الموجودة بالمعمل

| Serial Number | العدد | إسم الجهاز | م |
|---|---|---|---|
| | ا | ميكرو كنترول شركة أتمل | 1 |
| | 2 | وحدة ميكرو كنترول شركة ميكرو اتش | 2 |
| | 6 | PLC شركة بد | 3 |
| | 2 | PLC شركة LG | 4 |
| | | | 5 |
| | | | 6 |
| | | | |
| | | | |

Part
3

# 3: Laboratory Experimental List

## ثالثاً: قائمة بالتجارب التي تؤدي داخل المعمل

Part 4

# 4: Laboratory Beneficiaries

## رابعاً: الخدمات المجتمعية التي يؤديها المعمل:

Part 5

## 5: Laboratory Student Beneficiaries

## خامساً: الخدمات الطلابية التي يؤديها المعمل:

| | |
|---|---|
| عدد الطلاب المستفيدين من المعمل | في خلال مده اسبوع 200 طالب |
| الأقسام العلمية المستفيدة من المعمل | قسم هندسة الحاسبات والنظم |
| الفرق الدراسية المستفيدة من المعمل | إعدادي + 4 فرق القسم |
| المقررات الدراسية التي تستفيد من المعمل | برمجه للفرقة الإعدادي+ برامج الفرق المختلفه |
| الأنشطة الطلابية داخل المعمل | مشاريع القسم |
| **عدد طلاب الدراسات العليا المستفيدين من المعمل** | |
| عدد الرسائل العلمية التي تمت في المعمل | |
| عدد الدورات التدريبية التي تمت في المعمل | |
| المسابقات العملية التي شارك فيها طلاب من المستفيدين من المعمل | |

Part 6

# 6: Laboratory Experimental

## سادساً: اتجارب المعملية

بيانات المعمل الأساسية

Experiment No. (1)

# Siemens S300 Hardware Description

**الفرقة  المقرر عليها التجربة : الثالثة حاسبات**

**الفصل الدراسى : الأول**

**Objective**
- Understanding PLC Siemens-S300 CPU.
- The configuration of different modules.

الأدوات المطلوبة

**Tools**

- Siemens CPU S300.
- Simatic Step7 Software.
- USB to MPI Converter.

**Theory**

**SIEMENS S313 CPU**

**Signal Modules:**

- Digital input modules
  24 VDC – 120/230 VAC
- Digital output module
  24 VDC - Relay
- Analog input modules
  Voltage – Current – Thermocouple – Resistance
- Analog output modules
  Voltage - Current

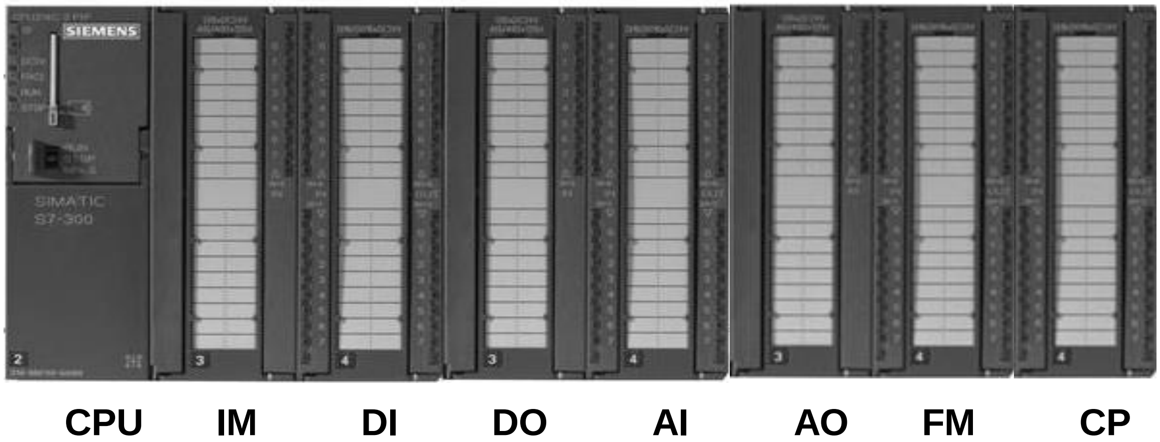



Fig.1 CPU 313C-2 DP PLC

**<u>CPU 313C-2 DP:</u>**

- The compact CPU with integrated digital I/Os
- PROFIBUS DP master/slave interface
- With process-related functions
- For tasks with special functions
- For the connection of standalone I/O devices

Note: Micro memory card required to operate the CPU.

**<u>Signal Modules:</u>**

Digital input modules
Digital output module
Analog input modules
Analog output modules

**<u>Memory Card</u>**

A slot is provided for a memory card. The memory card saves the program contents in the event of a power outage without the need for battery.

## The Question

What are the abbreviations of IM, DI, DO, AI, AO, FM and CP? What are the purpose of each one?

# Experiment No. (2)

# Planning subnets

**الفرقة  المقرر عليها التجربة : الثالثة حاسبات**

**الفصل الدراسى : الأول**

**Objective**

- To understand how PLC can be connected in a network and what are the different type of networks that can be used.

**Tools**

- Siemens CPU S300 RACK.
- Simatic Step7 Software.
- USB to MPI Converter.

**Theory**

Subnets available in SIMATIC for the various automation levels (process, cell, field and actuator/sensor level ):

- **Multi-Point Interface (MPI)**
- **PROFIBUS**
- **PROFINET (Industrial Ethernet)**
- **Point-to-point communication (PtP)**
- **Actuator/Sensor Interface (ASI)**

**The Question**

What are the subnets available for SIMATIC and what are the Difference between them?

**Experiment No. (3)**

**Expand No. of I/O of PLC S300**

**الفرقة  المقرر عليها التجربة : الثالثة حاسبات**

**الفصل الدراسى : الأول**

## Objective

- To understand how to expand no. of input/output modules.

## Tools

- Siemens CPU S300 RACK.
- Power and I/O Modules RACks.
- Simatic Step7 Software.
- USB to MPI Converter.

## Theory


Fig.2 CPU S300 RACK

The RACK is configured with the following modules:

Slot 1
Reserved for the Power supply.
Slot 2
Slot for the CPU (313C-2 DP).
Slot 3
Logically reserved for an interface module (IM)

Slots 4-11
Slot 4 is the first slot that can be used for I/O modules, communications processors (CP) or function modules (FM).

**The Question**

How much racks can be connected to a basic PLC S300 rack to expand I/O modules as maximum ? Draw the complete circuit diagram for I/O expansion?

## Experiment No. (4)

## Contacts and Coils

**الفرقة المقرر عليها التجربة : الثالثة حاسبات**

**الفصل الدراسى : الأول**

**Objective**

- To understand how to represent an input or an output to PLC.

**Tools**

- Siemens CPU S300 RACK.
- Simatic Step7 Software.
- USB to MPI Converter.

**Theory**

**Basics (tools) of the LD language: -**

## 1. Normally Open Contact (Address)

<address>

**---|   |---**

(Normally Open Contact) is closed when the bit value stored at the specified **<address>** is equal to "1". When the contact is closed, ladder rail power flows across the contact.
Otherwise, if the signal state at the specified **<address>** is "0", the contact is open. When the contact is open, power does not flow across the contact.

### 2. Normally Closed Contact (Address)

<address>

**---| / |---**

(Normally Closed Contact) is closed when the bit value stored at the specified **<address>** is equal to "0". When the contact is closed, ladder rail power flows across the contact.
Otherwise, if the signal state at the specified **<address>** is "1", the contact is opened. When the contact is opened, power does not flow across the contact.

### 3. Invert Power Flow

**---|NOT|---**

it is used to Invert Power Flow

### 4. Output Coil

<address>

**---(   )**

Output Coil works like a coil in a relay logic diagram. If there is power flow to the coil , the bit at location **<address>** is set to "1". If there is no power flow to the coil, the bit at location **<address>** is set to "0". An output coil can only be placed at the right end of a ladder rung. Multiple output elements (max. 16) are possible in parallel. A negated output can be created by using the ---|NOT|--- (invert power flow) element.

### 5. Midline Output

<address>

**---( # )---**

Midline Output is an intermediate assigning element which saves the power flow status to a specified **<address>**. The midline output element saves the logical result of the preceding branch elements. In series with other contacts, ---( # )--- is inserted like a contact. A ---( # )--- element may never be connected to the power rail or directly after a branch connection or at the end of a branch. A negated ---( # )--- can be created by using the ---|NOT|--- (invert power flow) element.

#### 6. Reset Coil

<address>

**---( R )**

Reset Coil is executed only if the preceding instructions is "1". If power flows to the coil , the specified **<address>** of the element is reset to "0". no power flow to the coil has no effect and the state of the element's specified address remains unchanged.

#### 7. Set Coil

<address>

**---( S )**

Set Coil is executed only if the preceding instructions is "1". the specified **<address>** of the element is set to "1". No power flow (0) has no effect and the current state of the element's specified address remains unchanged.

**8. Negative Edge Detection**

<address>

**---( N )**

Negative Edge Detection detects a signal change in the address from "1" to "0" . At this an output pulse is occurred.

**9. Positive Edge Detection**

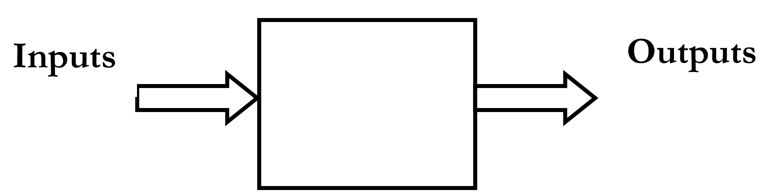
<address>

**---( P )---**

Positive Edge Detection detects a signal change in the address from "0" to "1" . At this an output pulse is occurred.

**10. Function blocks**

There are several blocks in the LD language. That each performs a particular task. We use each block according to the requirements.

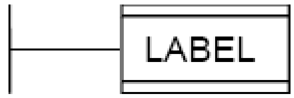



**11- Jump symbol**

<label name>

**---( JMP )**

**JMP** functions as an absolute jump when there is no other Ladder element between the left-hand power rail and the instruction. It is used when we want to jump to a particular rung in the program. Placed at end of rung. **JMP** functions as a conditional jump when the previous logic operation is "1". All instructions between the jump instruction and the label are not executed.




**<Label name> :** it is the rung label which we want to branch to it.

### The Program

Assume that we have two push button switches S1 and S2 and output for a motor . it is required to satisfy:

a. Pressing S1 and not pressing S2, then start the motor.
b. Pressing S2 or not Pressing S1, then start the motor.
c. Pressing S1 or pressing S2, then stop the motor.
d. Pressing S1 start the motor, and pressing S2 stops the motor.

## Experiment No. (5)

## A Simple Project

### Objective

- To understand how to begin programming simple projects.

### Tools

- Siemens CPU S300 RACK.
- Simatic Step7 Software.
- USB to MPI Converter.

### The Program

Drive the ladder diagram for the stamping and packing problem shown in fig. If the sensors a0, a1, b0, b1, c0 and c1 are placed to control the placement of the cylinders A, B and C. S1 to detect the object.

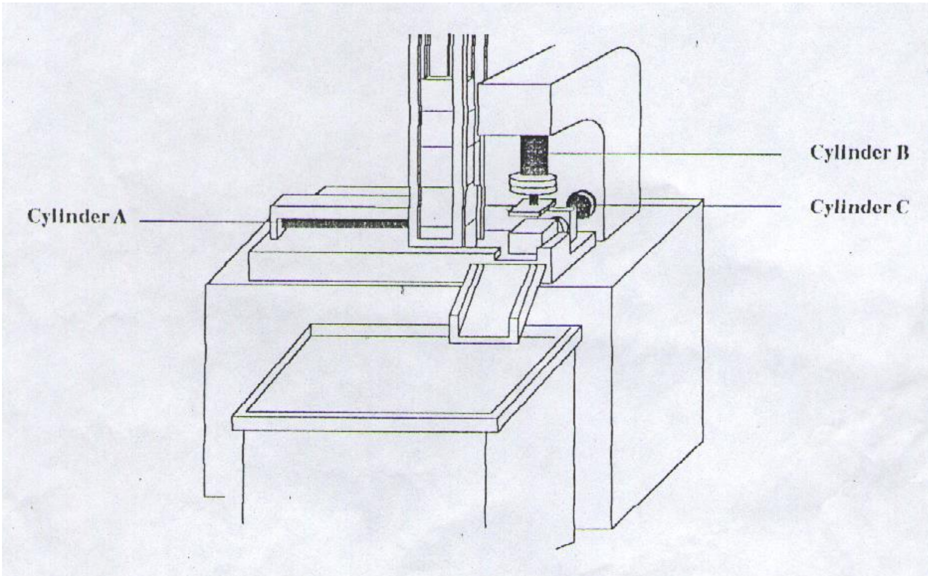



Fig.3 Stamping and Packaging Problem

# Experiment No. (6)

# Timer Function Blocks

**Objective**
-To understand the difference between timer blocks.
-To learn hot to use timer blocks in the program.

**Tools**

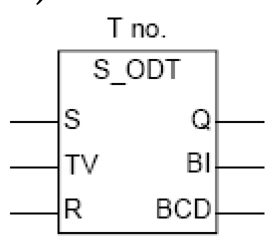
- Siemens CPU S300 RACK.
- Simatic Step7 Software.
- USB to MPI Converter.

**Theory**

| Timer | Description |
|---|---|
| **S_PULSE**<br>Pulse timer | The maximum time that the output signal remains at 1 is the same as the programmed time value t. The output signal stays at 1 for a shorter period if the input signal changes to 0. |
| **S_PEXT**<br>Extended pulse timer | The output signal remains at 1 for the programmed length of time, regardless of how long the input signal stays at 1. |
| **S_ODT**<br>On-delay timer | The output signal changes to 1 only when the programmed time has elapsed and the input signal is still 1. |
| **S_ODTS**<br>Retentive on-delay timer | The output signal changes from 0 to 1 only when the programmed time has elapsed, regardless of how long the input signal stays at 1. |
| **S_OFFDT**<br>Off-delay timer | The output signal changes to 1 when the input signal changes to 1 or while the timer is running. The time is started when the input signal changes from 1 to 0. |

Table1 Timer blocks

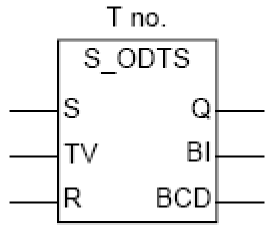
## 1. ON Delay Timer (S_ODT)




<u>Start</u>
The timer starts when the Start input "S" changes from "0" to "1". The timer starts with the time value specified at the Time Value "TV" for as long as the signal state at input "S" =1.

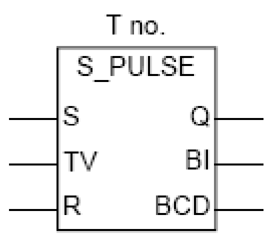
<u>Reset</u>
When the Reset input "R" changes from "0" to "1", the current time value and the time base are deleted and the output "Q" is reset.
<u>Output</u>
The signal at the "Q" output changes to "1" when the timer has expired without error and input "S" has signal state "1".
<u>Digital Outputs</u>
The current time value can be read as a binary number at the "BI" output and as a BCD number at the "BCD" output.

## 2. Stored ON Delay Timer (S_ODTS)




Start
The stored-on-delay timer starts when the "S" input changes from "0" to "1". The timer runs starting with the time value specified at input "TV" and continues to run even if the signal at input "S" changes back to "0" during that time.
Reset
When the reset input "R" changes from "0" to "1", the current time value and the time base are deleted and output "Q" is reset.
Output
The signal state at output "Q" changes to "1" when the timer has expired without error, regardless of whether the signal state at input "S" is still "1".

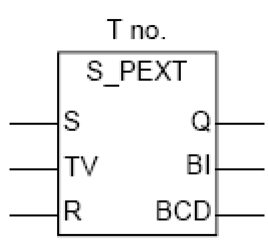
**3.Pulse Timer (S_PULSE)**




Start
The pulse timer starts when the "S" input changes from "0" to "1". Output "Q" is also set to "1".

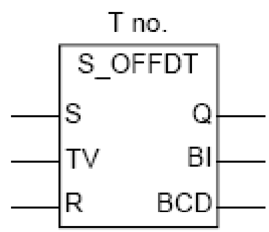
Reset
Output "Q" is reset when the timer has expired, or the start "S" signal changes from "1" to "0", or the reset input "R" has a signal state of "1".

**4. Extended Pulse Timer (S_PEXT)**



Start
The extended pulse timer starts when the "S" input changes from "0" to "1". Output "Q" is also set to "1".
The signal state at output "Q" remains at "1" even if the signal at the "S" input changes back to "0".
If the signal at the start input changes from "0" to "1" again while the timer is running, the timer is restarted
Reset
Output "Q" is reset when the timer has expired, or the reset input "R" has a signal state of "1".

**5. Off Delay Timer (S_OFFDT)**




Start
The off-delay timer starts when the "S" input changes from "1" to "0". When the timer has expired, the signal state at output "Q" changes to "0".
If the signal state at the "S" input changes from "0" to "1" while the timer is running, the timer stops. The next time the signal state at the "S" input changes from "1" to "0", it starts again from the beginning.

Reset
When the reset input "R" is "1", the current time value and the time base are deleted and output "Q" is reset.
If both inputs (S and R) have signal states of "1", output "Q" is not set until the dominant reset is deactivated.
Output
Output "Q" is activated when the "S" input changes from "0" to "1". If input "S" is deactivated, output "Q" continues to have signal state of "1" until the programmed time has expired.

**The Program**

By pressing Start pushbutton process will start.
Valve 1 will be opened for 10 seconds then it will be closed.

Valve 2 will be opened for 20 seconds then it will be closed.
Valve 3 will be opened for 10 seconds then it will be closed.
Mixer will run for 10 sec then it will be stopped.
Pump will run and valve 4 will be opened for 10 seconds.
Pressing Stop pushbutton will stop the process anytime.

## Experiment No. (7)

## Counter Function Blocks

**الفرقة  المقرر عليها التجربة : الثالثة حاسبات**

**الفصل الدراسى : الأول**

### Objective

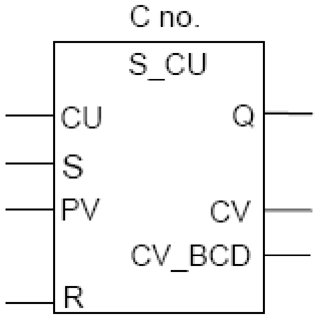
-To understand the difference between counter blocks.
-To learn hot to use counter blocks in the program.

### Tools

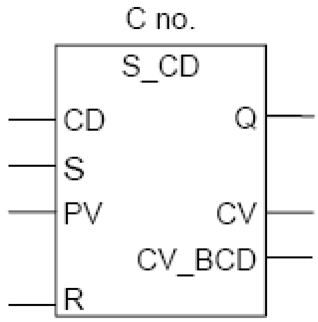
- Siemens CPU S300 RACK.
- Simatic Step7 Software.
- USB to MPI Converter.

## Theory

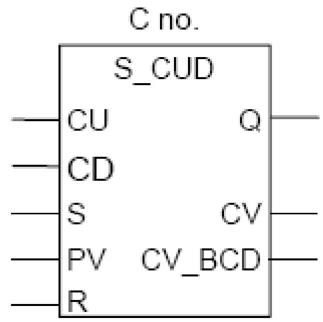
**S_CU Up**




**S_CD Down**




**S_CUD Up-Down**




<u>Count Up (CU)</u>

When the RLO at the "CU" input changes from "0" to "1", the counter's current value is incremented by 1 (upper limit = 999).

<u>Count Down (CD)</u>

When the RLO at the "CD" input changes from "0" to "1", the counter's current value is decremented by 1 (lower limit = 0).

<u>Set Counter (S)</u>

When the RLO at the "S" input changes from "0" to "1", the counter is set to the value at the "PV" input.

<u>Preset Value (PV)</u>

The preset value (0 to 999) is specified in BCD format at the "PV" input as: a constant (C#...) or a BCD format through a data interface.

<u>Reset Counter (R)</u>

When the RLO at the Reset changes from "0" to "1", the counter's value is set to zero. If the reset condition is fulfilled (stays "high"), the counter cannot be set and counting in either direction is not possible.

<u>CV/CV_BCD</u>

The counter value can be loaded as a binary number (CV) or BCD number (CV_BCD) into accumulator 1 and then transferred to other addresses.

<u>Output</u> (Q)

The signal state of the counter can be checked at output "Q":

Count = 0 -> output Q = 0

Count >< 0 -> output Q = 1

## The Program

Write a PLC program to control the counting between zero & 9, by using one input & one counter.

count up 0 → 9
count down 9 → 0

## Experiment No. (8)

## Practical Application

**الفرقة المقرر عليها التجربة : الثالثة حاسبات**

**الفصل الدراسى : الأول**

### Objective

-To understand the how PLC deal with the analog signals.
-To program the analog applications.

### Tools

- Siemens CPU S300 RACK.
- Simatic Step7 Software.
- USB to MPI Converter.

### The Program

The process gets a liquid from tank1 by opening V1 for 10 sec. And the same for tank 2. Then mixing the two liquids by the fan then after 5sec. Tank 3 will fill the bottle for 8 sec., and then this bottle is packed by cly. A and after 6 bottles cly. B moves them to another belt. From the given fig. Drive the ladder diagram. Assuming a start switch.

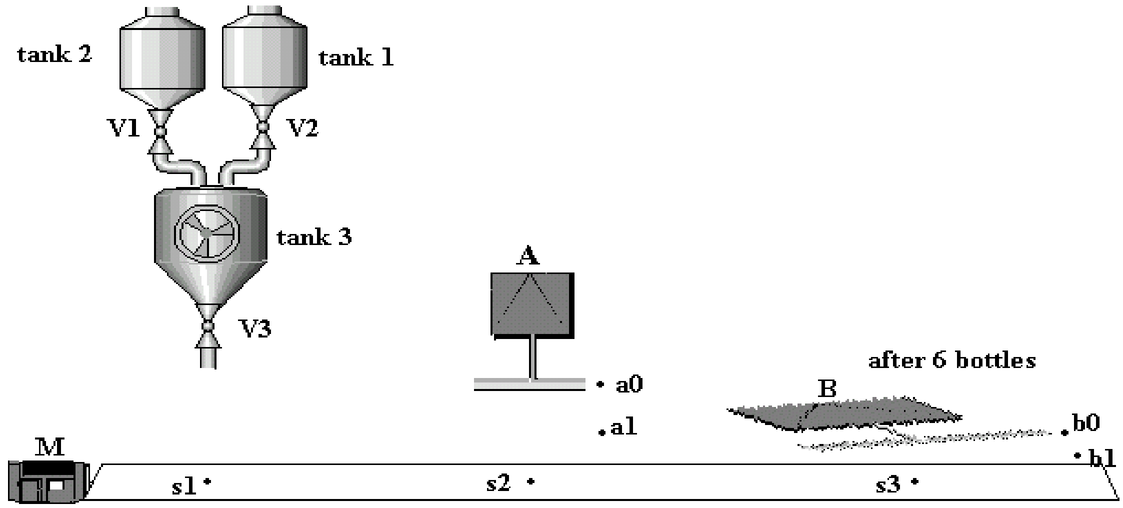



Fig.4 Mixing, Filling And Packaging Process

**Experiment No. (9)**

**Analog Application**

## Objective

-To begin programming a real world application.

## Tools

- Siemens CPU S300 RACK.
- Simatic Step7 Software.
- USB to MPI Converter.

## Theory

### Analog Input & Output Modules

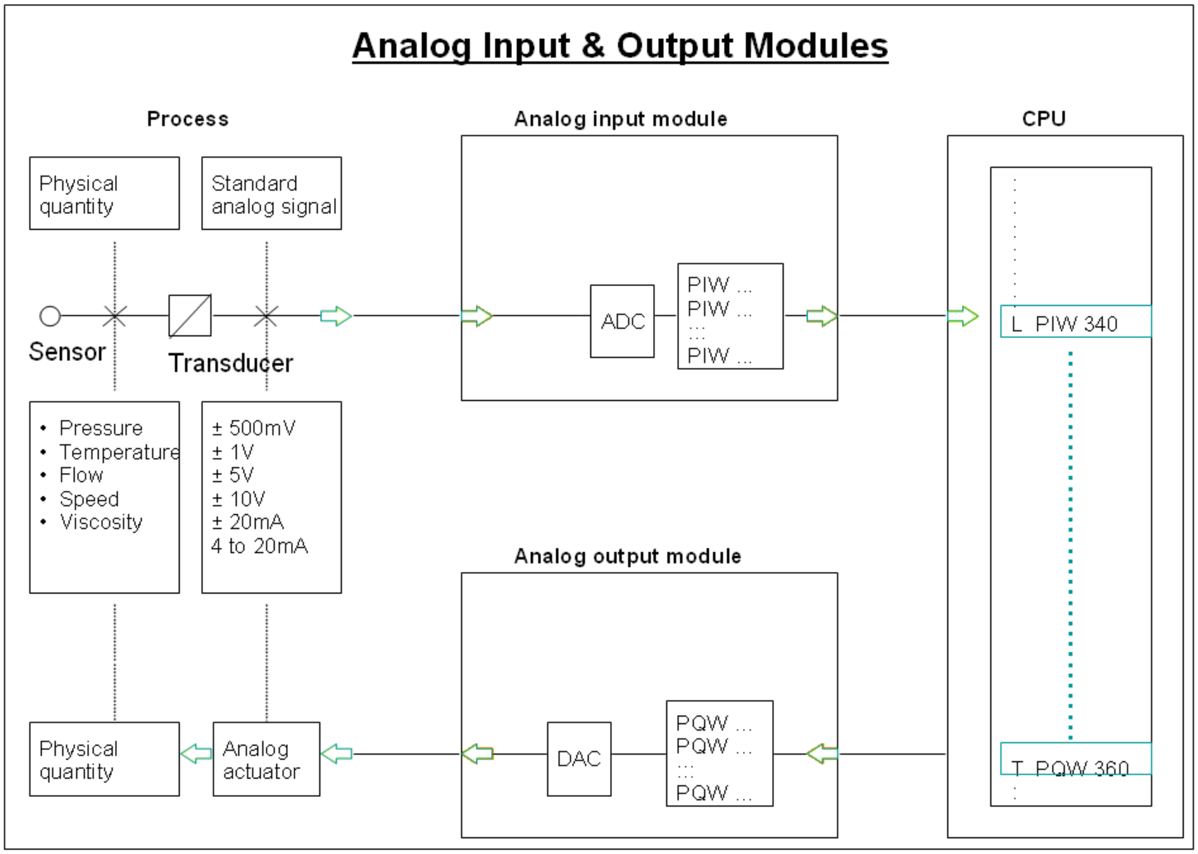
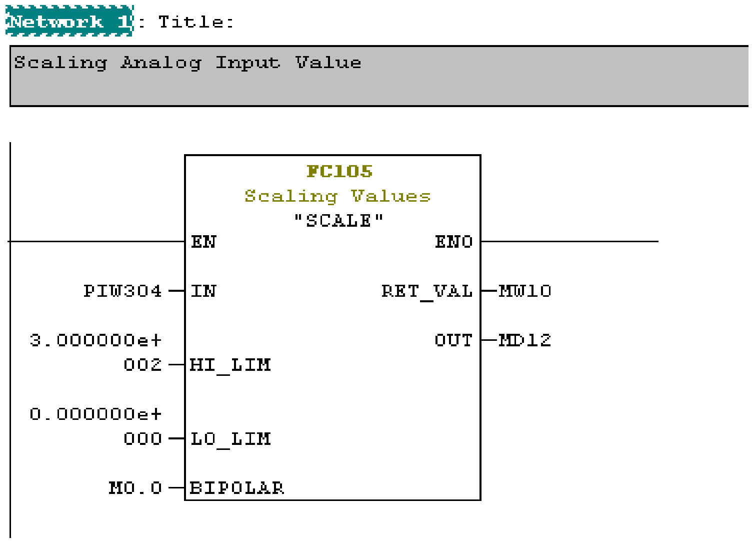



Fig.5 Analog I/O Modules

## Scaling Analog Input Values

### Scaling

The analog module encrypts the voltage range of -10V to +10V in the value range of -27648 to +27648. The conversion of this value range to the original physical size (such as 0 to 500l) is called scaling. The standard block FC 105 is used for scaling the analog value. FC 105 is supplied with the STEP 7 software in the "Standard Library" in the "TI-S7 Converting Blocks" S7 program.




The Reverse Function Called FC106

## Analog Value Representation of Different Measuring Ranges

| Range | Voltage such as: Meas.range ± 10V | Units | Current such as: Meas.range 4 to 20mA | Units | Resistance such as: Meas.range 0...300Ohm | Units | Temperature such as Pt100 Meas.range -200...+850°C | Units |
|---|---|---|---|---|---|---|---|---|
| Overflow | >= 11.759 | 32767 | >= 22.815 | 32767 | >=352.778 | 32767 | >= 1000.1 | 32767 |
| Overrange | 11.7589 : 10.0004 | 32511 : 27649 | 22.810 : 20.0005 | 32511 : 27649 | 352.767 : 300.011 | 32511 : 27649 | 1000.0 : 850.1 | 10000 : 8501 |
| Rated range | 10.00 7.50 : -7.5 -10.00 | 27648 20736 : -20736 -27648 | 20.000 16.000 : : 4.000 | 27648 20736 : : 0 | 300.000 225.000 : : 0.000 | 27648 20736 : : 0 | 850.0 : : : -200.0 | 8500 : : : -2000 |
| Underrange | - 10.0004 : - 11.759 | - 27649 : - 32512 | 3.9995 : 1.1852 | - 1 : - 4864 | Negative values not possible | - 1 : - 4864 | - 200.1 : - 243.0 | - 2001 : - 2430 |
| Underflow | <= - 11.76 | - 32768 | <= 1.1845 | - 32768 | | - 32768 | <= - 243.1 | - 32768 |

Fig.6 Analog Value Representation of Different Measuring Ranges

**The Program**

write the Ladder diagram of the following ***Oil Refinery***

-Pump is used to get the oil to refinery tank through path.

-S1 to detect that the tank is empty, S2 to detect that the tank is full.

-Heater is used to vaporize the contents of Oil at different temperature

degrees.

-Thermo analog sensor (0ºC-1000ºC) to measure temp. it is desired to open the valves one by one according to:

-V1 is opened in temperature range 100ºC-300ºC.

-V2 is opened in temperature range 500ºC-700ºC.

-V3 is opened in temperature range 750ºC-950ºC.

-The system must be stopped if the temperature over 1000ºC.

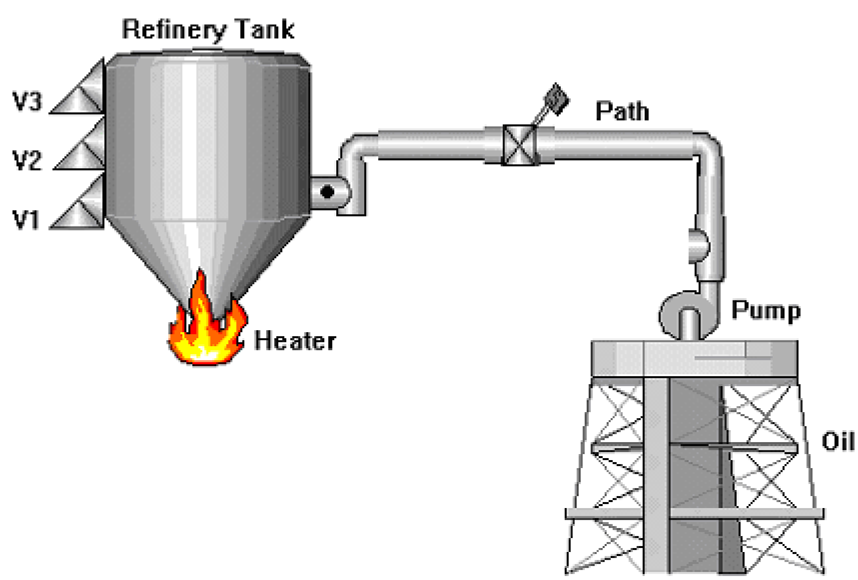


Fig.7 Oil Refinery Process