Gas-pressure regulator

SIMATIC
Industrial Software

Automation
with
natural gas

Totally
Integrated
Automation

47 - 1994

By Hans Ponck
June 1993

T.E.I. of Kavala
Greece

Stiftung
Hochschule
Mentors Greece: Mr. Stavridis Dimosthenis
Supervisor Belgium: Mr. Karasavas Climis
Class: Mr. Van Biervliet Jan
Year: E3KL
Period: 1998-'99
Lab of work: 31/03/1999-30/06/1999
Electrical installations

Gas-pressure regulator

SIMATIC Industrial Software
Automation and natural gas
Totally Integrated Automation

Performed by Hans Fonck, June 1999.

Technological institution of Kavala
Agios Loukas P.O. Box 1194
65404 Kavala
Greece

Katholieke Hogeschool St-Lieven
Depart Campus St-Annalaan
Sint-Annalaan 198
9300 Aalst
Ευγενιστώ πολύ / Thank you all!

For having this once in a lifetime opportunity I have to thank a lot a people you know. Without the approving and support of my parents, my family and my girlfriend I wouldn’t be here!

Also thanks to my school and the professors in Belgium and the Erasmus foundation who gave me this change. For the trust they had in me for representing a whole school. Here, in the T.E.I Kavala, I want to thank mister Kyranastasis and mister Stavridis for their help and advice, Mister Climis Karasavas for the advice he gave me during our meetings. And to Petros Sidiropoulos, who helped where and when he could.

Also a lot of people of Greece and Belgium I forgot to mention but who I needed to finish this work properly. My new friends, Greek and Erasmus people, that supported me with their feelings and sympathy!

But you did more than only helping, more than you will ever understand.

The only thing I can say: 'Thanks to you all, and may the future be kind for you with many nice surprises'.

Also I want to give you a little advice. An advice that makes the things easier for you and your environment ‘be yourself’ and ‘live for each other not for yourself’. At least it helps for me.

I know I can consider me a lucky man because this is a big step in attaining the full development of a person!

It was nice knowing you all and take care!

Hans Fonck

Kaho St.-Lieven
Table of contents

A general word

Contents

1 Gas pressure regulator
   1.1 Purpose
   1.2 Project
   1.3 About natural gas
      1.3.1 In general
      1.3.2 The use of gas in Belgium
   1.4 Safety control
      1.4.1 Emergency shutdown
      1.4.2 Process shutdown
      1.4.3 Fire and gas detection

2 S7-314 IFM
   2.1 Description
   2.2 Hardware
      2.2.1 Memory module
      2.2.2 Wiring
      2.2.3 Communication port
      2.2.4 Mode switch
   2.3 Software
      2.3.1 Objects
      2.3.2 Working with objects

3 Composing a project

4 The final work
   4.1 What and how
   4.2 The flowchart
   4.3 Explanation of the flowchart
   4.4 Difficulties
      4.4.1 First problem: FC105
      4.4.2 Second problem: wiring
      4.4.3 Third problem: the right way?
   4.5 The program
      4.5.1 OB1
      4.5.2 FC1
      4.5.3 FC2
      4.5.4 FC3
      4.5.5 FC4
   4.6 Remarks
   4.7 Suggestions

5 Conclusion

6 Bibliography
1. Gas pressure regulators.

Controlling a pressure drop system of natural gas in a pipeline by means of the S7-314 IFM.

Industrial Projects
and Technical Services

Oil & Gas Pipelines
1.1 Purpose.

The main part was learning how to work with the software used for configuring and programming a SIMATIC S7-300 version. Step 7 comprises the standard software and optional software packages which run under windows 95/98 or Windows NT. It’s the programming software for SIMATIC S7/M7 and therefore for your S7-300. STEP 7 provides you with the entire functionality required for configuring, programming, and assigning parameters to your S7-300. The programming software gives you effective support when solving your automation task.

Since it was all new for me, this learning took a great deal of my time here. The reading and studying of the reference bookwork was difficult because of the many new complicated contents. I was not used to the language that is used in the area of PLC and its software.

After this part I started to write small programs to test my knowledge on the PC and PLC! That until the destiny of my final project was given to me!

1.2 The project.

The idea is to control the valves in pipelines where natural gas flows through. These valves, needed to regulate the pressure and used as security-lock when there is a problem in the circuit, have to be controlled by the S7-314 IFM PLC. To put it shortly, we have to reduce high pressure to a pressure that is useful for the factory to work with.
This way of control and regulation you find in many applications. The same way and application is used when gas reaches our homes. The idea is equal everywhere but each area has its own specifications.

Procedure:

The natural gas comes under high pressure, 67 to 70 bar, and has to be reduced to a value around 19 bar. The pressure drop regulator executes this. We control two lines: the current line and a bypass. Each line has two valves, a pressure drop regulator and a low pressure gauge. The high pressure gauge is common for these lines.

Overview:

> Pressure-drop-regulator
In the case everything works normal and the system is error-free, the natural gas passes through the current line (line 1). From the moment of an error, a signal closes the two valves in the current line and restarts the program. But now with the bypass (line 2) as means of transportation. Line 2 is now the current line.

Normally, a mechanic fixes the problem in the current line. After finishing the work, he puts the reset button to restart the whole operation. Line 2 is back the bypass and line 1 the current line. Even when he forgets to strike the reset button there will be no problem. The same procedure will be executed as written above. And when the problem in line 1 still isn’t fixed by then, all the valves will be closed and the system remains non-active until all irregular things in the operation are solved.

The control of the valves works with a little power supply between 0 and 10 Volts. By reading the analog signal the PLC gives orders to the valves. The values from the high pressure are between 67 and 70 bars. Translated by the program this results in:
- \(0V < \text{HP} < 4V\) \(\Rightarrow\) HP is too low
- \(4V < \text{HP} < 8V\) \(\Rightarrow\) HP is OK
- \(8V < \text{HP} < 10V\) \(\Rightarrow\) HP is too high

The same procedure happens for the low pressure!

1.3 About natural gas:

1.3.1 In general

Natural Gas, also known as methane, is a colorless, odorless, fuel that burns cleaner than many other traditional fossil fuels. It is one of the most popular forms of energy today. It is used for heating, cooling, production of electricity and it finds many uses in industry. Increasingly, natural gas is being used in combination with other fuels to improve their environmental performance and decrease pollution.

Natural gas is produced, sometimes along with oil, by drilling into the Earth's crust where pockets of natural gas were trapped hundreds of thousands of years ago.
Once the gas is brought to the surface, it is refined to remove impurities, like water, other gasses, and sand. Then it is transmitted through large pipelines that span the continent. Factories and electric power plants may get gas directly from the pipeline using arrangements made through a marketer or supplier. Residential and smaller businesses generally buy gas from a local distribution company or utility. The distributor adds an odorant to the gas as a safety measure so that people will be able to tell if there is a gas leak, or if they forget to turn off an appliance.

Transportation of Natural Gas:

After raw gas from the wellhead is processed, it is moved into a pipeline system for transportation to an area where it will be sold. A pipeline company is a totally separate company from a producer or a distributor, although sometimes pipelines sell gas directly to large customers. The interstate pipeline system is massive, reliable, and efficient. Major investments in the pipeline system during the 1980's and early 1990's improved the system's capacity to areas in the Northeast, West Coast and Florida. However, the pipeline industry is still making improvements in capacity, efficiency and cost effectiveness, since transportation costs still make up a large portion of the consumer's price for natural gas.
Most sections of pipeline are made of steel piping, measuring anywhere from 20 to 42 inches in diameter. When natural gas is moved through a pipeline, it is transmitted at higher pressures (from 200 to 1500 psi) to reduce the volume of the gas, and provide a pushing force to propel the gas through the pipe.

In order to maintain the level of pressure required to move the large volumes of gas through a pipeline, the gas needs to be compressed periodically as it moves through the pipeline. This requires pipelines to install compressor stations every about every 100 miles along the pipeline.

Most of these compressors are classified as reciprocating compressors, which means that they are powered by a very small portion of the natural gas that flows through the pipeline. These compressors are efficient and safe, their only drawback being that they tend to be quite large. There are over 8,000 gas compressing stations along gas pipelines, with a combined output capability of over 20 million horsepower.

A problem:

One of the classic environmental problems with any sort of energy is that a portion of the energy is lost in transporting it from its source to its destination. Gas transportation is very efficient in this respect, compared to other energy resources. Only about 3 percent of the gas energy that is transported is lost in the process. When considering the efficiency of an energy resource from start to finish, gas appears even more efficient. For example, the use of natural gas is much more efficient than using electricity. Electricity delivers less than 30% of the natural energy to your home because so much energy is lost in generating electricity. Over 70% of the natural energy used to generate electricity is lost during electric generation and powerline transmission to your home. Natural gas delivery to your home is over 90% efficient.

The U.S. gas transmission system is composed of over 300,000 miles of piping, not including local distribution lines.

These pipelines need to be monitored 24 hours a day and 365 days a year. In order to keep accurate, constant information on sections of pipeline, pipeline companies use 'supervisory control and data acquisition systems' (SCADA systems diagram). These are computerized systems that allow pipeline operators to acquire information from remote sections of pipeline, and also control the flow of gas at remote locations by using computers that are linked to satellite communication and telephone communication systems.
SCADA systems allow not only the pipeline operators to obtain timely information, but they also allow producers to have access to some of the same information so that they can purchase distribution services according to the current volume of gas in a pipeline.

The information that is provided to those shipping gas on pipelines is posted on electronic bulletin boards (EBBs), which can be accessed by users in order to purchase transportation service, check on billing, or arrange storage of gas that has been transported through a pipeline. The Federal Energy Regulation Commission has begun to require pipeline companies to post information about pipeline utilization on such EBBs, and with the recent unbundling of pipeline services, it is beneficial to a pipeline company to provide such information so that its capacity can be used efficiently.

Another method that pipeline companies use to maintain their pipelines is the use of intelligent PIGs (intelligent robotic inspection devices). Not like your typical farm animal, these PIGs are used to inspect pipeline interior walls for corrosion and defects, measure the interior diameter of a section of pipe, and to remove accumulated debris from a section of pipeline.
As a PIG travels through a pipeline, it takes thousands of measurements with its accurate sensors that can later be analyzed and modeled by computers for a pipeline to show possible problems. Although pipelines use cathodic protection for many newer sections of their pipelines, they still encounter corrosion problems that weaken some parts of the pipeline. Magnetic-flux leakage PIGs are used to detect metal loss in pipeline walls, locating potential problems without the cost of using other methods.

Overall, delivering natural gas is among the safest means of distributing energy to customers. Much of this is due to the fact that the transmission system is fixed, and buried underground. Statistical data collected by the National Transportation Safety Board indicate that energy transportation by rail or truck represents a much higher safety risk than transportation through a pipeline. According to data from the U.S. Department of Transportation (DOT), natural gas and petroleum liquids pipelines are the safest method of transporting energy. For example, electric current is responsible for more than 100 deaths a year during its transmission to the customers. In contrast, in 1993, the most recent year for which data is available, only 14 pipeline accident fatalities were reported, according to DOT's National Transportation Safety Board (NTSB).

**Compression:**

Natural gas is compressed during transportation and storage. The standard pressure that gas volumes are measured at is 14.7 Pounds per Square inch (psi). When being transported through pipelines, and when being stored, gas is compressed to save space. Pipelines have compressing stations installed along the line (one about every 100 miles) to ensure that the gas pressure is held high while the gas is being transported. Current pipelines can compress natural gas to nearly 1500 psi, but most tend to operate at closer to 1000 psi.
**Electrochemical Sensors for Toxic Gas:**

Electrochemical sensors monitor specific toxic gases in the low Parts Per Million (PPM) range. Control Instruments’ electrochemical sensors offer high selectivity and accuracy to specific compounds, low maintenance, excellent repeatability and long-term stability. An on-board heater protects the cell and extends its useful operating range. Electrochemical sensors are available for the following gases:

**Carbon Monoxide**
Ranges: 0-100 or 0-1,000 PPM.
Applications: garages, tunnels, factories, steel mills, chemical processing, engine test cells, warehouses, pump stations and vaults.

**Nitrogen Dioxide**
Ranges: 0-10, 0-50, 0-100 PPM.
Applications: electrical generating stations, industrial furnaces and boilers, oil and gas industry, and mining industry.

**Sulfur Dioxide**
Ranges: 0-10, 0-50, 0-100 PPM.
Applications: pulping processes, sulfur processing plants, waste water treatment plants, battery manufacturing, copper smelters, industrial boilers, and petroleum refining.
The automation with Siemens SIMATIC S7

Hydrogen
Ranges: 0-500, 0-1,000, 0-2,000 PPM
Applications: semiconductor processes, hydrogen processing plants, petroleum industry, laboratory safety and battery rooms.

Hydrogen Sulfide
Ranges: 0-25 or 0-50 PPM
Applications: gas-processing plants, sulfur recovery plants, sewage plants, petrochemical and chemical plants.

1.3.2 The use of gas in Belgium.

This information I found on the website of Electrabel, a big company in Belgium that supports us with power supply.

How does natural gas reach you, the customer?

Natural gas is one of the energy sources of the future. Natural gas is clean energy: when burned it releases low levels of acid and carbon dioxide (CO₂). Not only that, it can be used in a number of ways and its supply is secure in Belgium.

Belgium has no natural gas reserves. Distirgas imports, stores and transmits natural gas for Belgium. It also sells it to power stations, heavy industry and the mixed intermunicipal companies. The mixed Intermunicipal companies then redistribute it to private individuals, small and medium-sized businesses and the tertiary sector.
Electrabel receives its natural gas via city gate stations, where the pressure is brought down to 15 bar. A number of large companies use gas at this pressure. The gas is then transmitted to the municipalities through the medium-pressure network (at 15 bar) linking the various pressures reducing stations and thus providing a continuous supply. The gas is actually distributed at a medium pressure of 5 bar, or at a low pressure of 100 or 25 millibar.

Electrabel has decentralized its distribution structure so that it is in the best possible position to meet the needs and expectations of its partners and customers at both regional and municipal level. The mixed intermunicipal companies, in which Electrabel is a partner, account for 89% of natural gas distribution in Belgium.
Today, more than 420 out of Belgium's 589 municipalities are supplied with natural gas, and more than 45% of homes are heated with this clean energy.

If you would like to know whether natural gas is available in your municipality and who distributes it, why don't we help you find your mixed intermunicipal company?

1.4 Safety Control:

Some information about several possibilities the market has for having a safe installation where the changes for injuries are minimized as much as possible.

1.4.1 Emergency Shutdown

The ESD system is based upon the powerful Fail Safe system SIMATIC S7-417 H/F. The fail-safe functions are achieved by utilizing fail-safe I/O and software. The basic system unit and connections to I/O are approved according to SIL 2 or 3 by TUV. Implementation of application for the shutdown part is done according to IEC 61508. The central unit is a redundant unit, SIMATIC S7-417H with redundant connections to all I/O racks.
All shutdown input- and output signals are hardwired to the unit. These signals are redundant, fail to safe, with monitoring. Due to the critical functions of the ESD system, special considerations are taken for the output signals. Output signals from the ESD will be duplicated with digital inputs for test. A program in the ESD will perform a cyclic test of the outputs to ensure reliable action when required.

Main operator interface is the ESD Matrix, but all detailed information shall be available on the ESD pictures on the operator stations, or optional, on large screens.

The unit performs shut down of all process parts, depressurization of the process, and performs all major electrical isolations.

1.4.2 Process Shutdown

The PSD system utilizes the redundant SIMATIC S7-400H system, certified to SIL 2 by TUV. Fail safe I/O and software will be used for fail safe requirements (i.e. shutdown inputs and outputs).

PSD levels for all process equipment are handled in PSD nodes, while the utility nodes can perform PSD- functions for utility equipment. All inputs are periodically scanned and shutdown is performed immediately after receiving a PSD input.

PSD logic and status can be observed in dedicated pictures on the OS. Typical functions in PSD system are:

- PSD digital input monitoring
- PSD level monitoring and control
- PSD digital output actions

1.4.3 Fire & Gas Detection

The F&G system are designed to monitor geographical parts of the plant. In addition to fire and gas detection, fire fighting and initiation of PA alarms, the F&G system handle all fire dampers with feedback, and local electrical isolation initiated from F&G only.
The F&G nodes can be singular redundant units with processor S7-417 (H) with single I/O

I/O in zone 2/1 areas is gated through distributed I/O stations ET200M, via serial link and/or Profinbus from fire stations.

The F&G systems are designed to monitor geographical parts of the plant. In addition to fire and gas detection, fire fighting and initiation of PA alarms, the F&G system handle all fire dampers with feedback, and local electrical isolation initiated from F&G only.

Fire fighting outputs are monitored normally open, with block facility.

Outputs for dampers and electrical isolation are fail to safe with override.

Light and sound are via the PA system.

The operator interface is split between the F&G mimic/matrix, showing the main alarms, and operator stations, showing all details. Manual release of fire fighting and block of automatic actions are done from the F&G matrix, while other operations are done from the operator stations.

Note:

Before the start of explaining my program I will give you some more information about the kind of PLC I’m going to use in my application.

You need the basic knowledge to have the possibility to understand the way of working of the CPU 314 IFM.

Then, you will have an idea of the things you can accomplish with such an automation instrument. It is a lot, it’s a pity that I only can show you a very small part of the possibilities you have.
What is the S7-314 IFM?
2.1 General description:

The custom solution for extremely fast processes or automation tasks requiring additional data processing capabilities. That is the SIMATIC S7-300: it can be related with these 3 words.

- **Powerful**
  - High computing performance, complete instruction set, multiple point interface (MPI) and networking capability via SIMATIC NET LANs make this PLC extremely powerful.
  - Built-in functions, comprehensive diagnostic capabilities, password protection, convenient connection system and unlimited plug-in module configurations facilitate handling.

- **High speed**
  - Extremely fast instruction processing achieves shorter cycle times.

- **Versatile**
  - High performance modules and six CPUs for a wide variety of requirements provide a suitable solution for every application.
  - Modular expandability with up to 3 expansion racks (ERs), extremely high packaging density, backplane bus integrated into the modules, and prefabricated connection system (TOP Connect) reduce space requirements and costs.
  - Family interface for connection to components of the SIMATIC family, built-in HMI (human machine interface) services, user-friendly STEP 7 basic software make the SIMATIC S7-300 into a universal mini PLC.
Now you have a general idea what this programmable controller is all about. But using this PLC is still something else. We are going to take a look at all the things we have to do and the requirements we need before we start working with the controller. First, I will write some words about the hardware and after the software! But to run an automation task, we need the whole concept, hardware and software.

2.2 Hardware components

The central unit of the S7-300 CPU (central processing unit) consists of a main memory module, a mode switch, a communication port, a battery, and wiring terminals.

2.2.1 Memory module

Memory of the S7 CPU has three basic areas:

- The load memory is used for programs without symbolic address assignments or comments (these remain in the memory of the programming device). The load memory can be either RAM or EPROM. Blocks identified as being not relevant to the running of your program are all located in the load memory.

- The work memory (integrated RAM) contains the parts of the S7 program relevant for running your program. The program is executed only in the work memory and system memory areas.

- The system memory (RAM) contains the memory elements provided by every CPU for the user program, such as the process image input and output tables, bit memory, timers, and counters. The system memory also contains the block stack and interrupts stack. In addition to the areas above, the system memory of the CPU also provides temporary memory (local data stack) that contains temporary data for a block when it is called. This data only remains valid as long as the block is active.
The memory of the S7 CPUs is divided into address areas. Using instructions in your program, you address the data areas directly in the corresponding address areas. To find out which address areas are available on your CPU, look to the CPU descriptions of the books of Siemens.

So, the CPU 314 has an optional FEPROM (Flash EPROM) memory card for non-volatile storage of your program. The memory card is not required for the programmable logic control to operate. The memory card provides the capability to field upgrade of a program (bug fixes, additional functionality, etc.) without having to use a programming device. This feature is especially useful for OEMs. In order to program the memory card, you must use either a PG or an EPROM programmer.

The programming of the memory card is accomplished by attaching an EPROM Programmer to the CPU. You can then download the program from the PLC into the memory card. It copies the data: user program, CPU programmable Parameters and the signal module parameters, this when the card is loaded.

When you power up the programmable logic controller with the memory card installed, the contents of the memory card are copied to the internal non-volatile memory. If you power up the PLC with a blank card, the controller will indicate a fault condition. You can install or remove the memory card while the unit is powered up.

You can find this card on front of the CPU and it is keyed for proper installation.

2.2.2 Wiring.

For wiring the CPU you should observe the assembly guidelines which are described in the manuals. S7-300 Programmable controller =>1 hardware and installation and 2 module specifications. You can wire the PS 307 power supply to this CPU 314 IFM via the connecting collar enclosed. Look at the picture below and you can see this is very easy!
The CPU has the following power supply terminals:

- Functional ground
- M (24 VDC)
- L+ (24 VDC)
- M (24 VDC)

2.2.3 Communication port

This PLC has an MPI communication port located under the lower access cover on the front of the CPU. The port uses an RS 485 9-pin sub D connector that allows you to attach either a programming or an interconnecting cable. The baud rate for programming is set at 187,500 baud.

If you want more information about the MPI, search for it in the topic help.
2.2.4 Mode switch

The key to switch from one mode to another is located on front of the CPU. With the tool you choose the operating mode you want! But it is not necessary to use the key because everything can happen with your PC. You can give any demand with the PC and he transfers it to the PLC. There are four different modes each connected with a led for knowing in what state the CPU is at the moment:

- **RUN-P (program)**
  The key can't be taken out now and the controller scans the user program.
  The program can be read out of the CPU. Or can be loaded into the CPU, the software can control the mode of the PLC into run or stop.

- **RUN**: The CPU scans the user program. The mode cannot be changed now, this to protect the machine and persons who work with the machine.

- **STOP**: The user program is not scanned now but the key can be removed! This to prevent from changing to another mode.

- **MRES**: The reset button to reset the whole program that was running at the time.

There are also some other leds that can burn but they do not have to be explained, you can read on the CPU itself next to the led was is wrong.

These were the most important subjects of the hardware of the PLC 314 IFM, so now we can take a look at the software, because the one can not without the other and visa versa.
2.3 Software.

This is in fact the same as the S7- SIMATIC program, the software we use to write a program in. I will explain the main steps in creating and writing a program later.

STEP 7 is the basic programming and configuration software for SIMATIC. It is made up of a series of applications, each of which does a specific job within the scope of programming an automation task, such as:

- Configuring and assigning parameters to the hardware
- Creating and debugging user programs
- Configuring networks and connections

A range of optional packages, for example, additional programming language, can extend the basic package packages such as SCL, S7-GRAPH, or HiGraph.

The graphic user interface provided for these tasks is known as the SIMATIC Manager.
The SIMATIC Manager collects all the data and the settings necessary for an automation task together in a project. Within this project the data are structured according to their function and represented as objects.

2.3.1 Objects

In a similar way to the directory structure used in the Windows Explorer containing folders and files, a STEP 7 project is divided up into containers and objects. Objects that can contain other containers and objects are known as containers, for example, an S7 program that contains the containers "Blocks" and "Source files" and the object "Symbols".

2.3.2 Working with Objects

The different types of object are linked directly in the SIMATIC Manager to the application in which they are required for processing. This means you do not have to remember which application you have to start in order to edit a specific object. You simply have to know what you have to do. We recommend that you read the STEP 7 User Manual before you start work with STEP 7 to give yourself an overview of the basic principles, the variants, and the recommended procedures. If you are developing an S7 or M7 user program for the first time, the programming manuals "Program Design" for S7-300/S7-400 and for M7-300/M7-400 will provide the information you require about the structure and the elements of a user program.
3. Composing a project
First

The first thing you need to do is to click on the icon of the S7-SIMATIC on your desktop. This in the case the software is already installed on your PC.

⇒ Double-click

Second

Then the following window appears on your screen and you are ready to work with the software of the S7.
We see now that the SIMATIC wizard appears to create the project. With the wizard you choose the kind of PLC you work with and select for the language you want to write the project in.

I took the ladder language because it was the kind of language in where I could work the best way. Maybe also because the view of seeing a program in ladder is familiar with reading an electrical drawing. I could more or less visualize the way of working of the program. The other two languages are more abstract to use, you have to know the use very well. With ladder, for me, there is more relation between the different orders and during the writing of the program you better the evolution.

Third

Then you choose the name of the project and you can start with writing the project.

If you don’t want to use the wizard you can also use the SIMATIC manager itself, below you find the two ways of working:

A. With the wizard:
The automation with Siemens SIMATIC S7

B. With SIMATIC manager:

Fourth

After creating the project you can with the tool 'insert' any station, program or function that you want. You do not have to do it at the beginning of your work, during you write the program you simply add the missing function(s) or block(s). Only the station and the program, in my case the S7-program, these two applications have to be defined at the beginning.

A. Insert the station or the program:
B. Inserting a block or a function:

Now we know the basic steps of the SIMATIC step7 manager. We can start writing any program we want. With a little help of the books, a lot of logic thinking and discuss the solutions or problems with a person who can help you, you can atomize any working system in a factory that you want!
And finally

After writing the program you have the possibility to test the program on the
without using the machines you are going to control. This is a theme, a kind of
surplus from the SIMATIC manager with a great advantage.
Running the program without the change you damage the machine(s) or the
person(s) that work near the application.
The way of testing is very simple and I will show them in a few steps.
When you run the program with the testing modulation you see what part of the
machine is activated, what values the CPU is reading in and when you have a
problem, where the program went wrong.
Testing the program has in the manager system an other name ‘DEBUG’ and ‘monitoring’. With these two themes you test a whole program. There are a few things you should now about debugging your program:

In order to debug your user programs, the relevant program parts must first be downloaded to the CPU and processed there. We strongly recommend you do not call the whole program and debug it in one, but debug the program components one by one.

You can:

- Display the actual values of individual variables of a user program or a CPU on the programming device or PC: Monitor Variables
- Assign fixed values to particular variables of a user program or a CPU: Modify Variables
- Assign fixed values to individual peripheral outputs of a user program or a CPU in STOP mode: Enable Peripheral Outputs
- Assign individual variables of a user program or a CPU a fixed value that cannot be overwritten by the user program: Force Variables

**CAUTION:**

Testing a program while a process is running can lead to serious damage to property or persons if errors occur in the function or in the program. Ensure that no dangerous situations can occur before you execute this function.

The variables you want to display or assign fixed values to be collected in variable tables (VAT). These variable tables can be stored in the project structure beneath the S7 program or M7 program in the "Blocks" container.

You can monitor and modify the user program in the CPU either with or without a project structure.

*Showing the way of working for testing:*

A little remark first, this has to happen when your are online with your program!
First step: debug $\Rightarrow$ call environment

Second step: monitor $\Rightarrow$ and you see what is activated or not, in what color and the size of the lines you want the monitoring is up to your own preference!
The result of monitoring
4. The final work
4.1 What and how?

The description of what I had to do, that you know already but the way that I accomplished the work is a way full of challenges and surprises. This way was long, sometimes I didn’t see light in the ‘tunnel’.

As you know I started to program with the S7-212 from Siemens SIMATIC. The smaller brother of the 300 version. I moved fast in the knowledge of this very good workable CPU, but after a few weeks mister Stravidis told I would need analog inputs to complete my program. Since the 212 CPU in the lab has no extension module for analog inputs I had to start working with the 314 version. This CPU has four analog inputs and one analog output. The way of programming is the same, more or less. But the use of the program manager is a lot more complicated than the small one. It has a lot more possibilities. In the beginning there were nothing else than problems, even running the simplest program was not possible!

There were some misunderstandings, between the books and myself, wrong wiring of the in- and outputs and a wrong configuration of the hardware! These things made the start of working with the S7-314 IFM laborious. After a few weeks of reading, asking questions, searching for answers, and some phone calls to Siemens in Thessaloniki, made me solve the first problems I had to beat.

During that time I started already with writing the paperwork for the project, but than someone decided to delete all the files I had! Everything, in every directory, two months of work disappeared in a glance!

Fate decided I had to start all over again. Two days I was a total mess, full with stress, terrible! I had to restart the work if I wanted to get my degree this year, so it became long working days of 14 to 16 hours a day! I hope for a little understanding because of the not always even nice layout, but I was a little bit running out of time.

In the following parts I will explain you the whole program, how I made progress, the difficulties and where the problems were situated, and as you will read, there were a lot of problems I had to deal with.
4.2 The flowchart.

Together with the help of mister Karasavas we compiled all our ideas together in a diagram and after a few weeks the final result was there. Mister Karasavas helped me very well in that part because I had this ‘accident’ with my PC. I added my point of view and made some changes, I will let you see the different steps in the flowcharts.

Flowchart 1:

Flowchart 2:
Flowchart 3:

This is the final drawing and because of its importance I created it on an A3 paper to have a better overview.
4.3 Explanation of the flowchart:

First we have to know what in- and output is what device to know what we are controlling at a certain time.

Outputs:
1. Q124 0 → alarm HP-led, HP > HL
2. Q124 1 → alarm HP-led, HP < LL
3. Q124 2 → alarm LP-led, LP > HL
4. Q124 3 → alarm LP-led, LP < LL
5. Q124 4 → Electrical valve HP 1 (=EV LP 1)
6. Q124 5 → Electrical valve HP 2 (=EV HP 2)
7. Q124 6 → Electrical valve LP 1 (=EV HP 1)
8. Q124 7 → Electrical valve LP 2 (=EV HP 2)
9. Q125 0 → signal EV HP 1 is completely open
10. Q125 1 → signal EV HP 2 is completely open

Inputs:
- 3 analog input signals simulated with a signal generator.
- 1124 0 → general start
- 1124 1 → general stop
- 1124 2 → switch that detects that the EV 1 is completely open
- 1124 3 → switch that detects that the EV 2 is completely open
- 1124 4 → general reset

Flowchart in words:

START

Identification of current line and bypass.
EV LP 1 = Q124 4
EV LP 2 = Q124 5
EV HP 1 = Q124 6
EV HP 2 = Q124 7
Reading the analog input = reading the HP

HP > HL ⇒ alarm Q124 0
HP < LL ⇒ alarm Q124 1

HP not OK ⇒ return for reading the pressure until HP = OK

Is HP = OK

EV LP 1 OPEN = set Q124 4
HP > HL ⇒ alarm Q124 0
HP < LL ⇒ alarm Q124 1
This remains

Is EV LP 1 OPEN ⇒ start reading analog input LP

Is LP < HL ⇒ OPEN EV HP 1 = set Q124 6

Is Q124 6 set and detection of valve completely open
(H124 2 for CL and H124 3 for BP)
The automation with Siemens SIMATIC S7

Set Q125.0 \(\Rightarrow\) EV HP is completely OPEN
(for BP set Q125.1)

is Q125.0 not set \(\Rightarrow\) return reading low pressure
   go on with reading the L and HP

LP < LL \(\Rightarrow\) alarm set Q124.3
LP > LL and > HL \(\Rightarrow\) alarm set Q124.2
   LP = OK and HP > HL \(\Rightarrow\) CLOSE EV HP 1 and EV LP 1
   LP = OK and HP < HL \(\Rightarrow\) back reading LP after opening EV HP

Is Q124.3 set and HP > LL \(\Rightarrow\) problem in working line switch to other line
   Is Q124.3 set and HP < LL \(\Rightarrow\) CLOSE working valves
   Is Q124.2 set \(\Rightarrow\) switch to other working line

Switch to other line \(\Rightarrow\) CLOSE working valves

Restart the program but with the other valves, and so the other line, as working!
Now is the CL = BP and BP = CL or visa versa
4.4 Difficulties I had in writing the program:

First there were troubles in understanding the use of function 105. This ‘scale’ function is used for reading analog inputs. But because there is practically no information in all the lecture books of the SIMATIC manager about the function, it took at least a week for understanding how the application worked. Because of these difficulties I will explain the function a little more.

4.4.1 First problem: ‘Scale’ function 105:

You can find the function in the library by doing the following steps:
Description

The SCALE function takes an integer value (IN) and converts it to a real value in engineering units scaled between a low and a high limit (LO_LIM and HI_LIM). The result is written in OUT. The SCALE function uses the equation:

\[ \text{OUT} = \left\lfloor \text{FLOAT} (\text{IN}) - \frac{\text{K1}}{\text{K2}-\text{K1}} \cdot (\text{HI_LIM}-\text{LO_LIM}) \right\rfloor + \text{LO_LIM} \]

The constants K1 and K2 are set based upon whether the input value is BIPOLAR or UNIPOLAR:

- **BIPOLAR**: The input integer value is assumed to be between -27648 and 27648, therefore, K1 = -27648.0 and K2 = +27648.0
- **UNIPOLAR**: The input integer value is assumed to be between 0 and 27648, therefore, K1 = 0.0 and K2 = +27648.0

If the input integer value is outside the specified range for its type (BIPOLAR or UNIPOLAR), the output (OUT) is clamped to the nearer of either LO_LIM or HI_LIM and an error is returned.

Reverse scaling can be obtained by programming LO_LIM > HI_LIM. With reverse scaling, the value of the output decreases as the value of the input increases.

**FC105 Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EN</strong></td>
<td>Input =&gt; Enable input with signal state of 1 activates the box</td>
</tr>
<tr>
<td><strong>ENO</strong></td>
<td>Output =&gt; Enable output has a signal state of 1 if the function is executed without error</td>
</tr>
<tr>
<td><strong>IN</strong></td>
<td>Input =&gt; The input value to be scaled to a real value in engineering units</td>
</tr>
<tr>
<td><strong>HI_LIM</strong></td>
<td>Input =&gt; Upper limit in engineering units</td>
</tr>
<tr>
<td><strong>LO_LIM</strong></td>
<td>Input =&gt; Lower limit in engineering units</td>
</tr>
</tbody>
</table>
BIPOLAR = Input ⇒ A signal state of 1 indicates that the input value is bipolar A signal state of 0 indicates that the input value is unipolar.

OUT = Output ⇒ Result of the scale conversion.

RET_VAL = Output ⇒ Returns a value of W#16#0000 if the instruction executes without error. See Error Information for values other than W#16#0000. Error Information

If the input integer value is outside the specified range for its type (BIPOLAR or UNIPOLAR), the output (OUT) is clamped to the nearer of either LO_LIM or HI_LIM and an error is returned. The signal state of ENO is set to 0 and RET_VAL is equal to W#16#0008.

Example:

We see we read the analog value in at address PIW128, the HI = 10 and LL = 0, the value we read out and that we use in the program is MD0. So is the switch I124 0 closed and we read an analog value between 0 and 10 we have an output in Q124 0.
4.4.2 Second problem.

The second problem was the addressing and wiring of the analog inputs. Here you find a diagram of the way this part of the CPU is constructed:

*Diagram was sent by fax by Siemens Germany.*

---

**Diagram: Anschlussbild der CPU 314 IFM**

- **Bild 10-10** zeigt das Anschlussbild der CPU 314 IFM.
- Zur Verdrahtung der integrierten Ein- und Ausgänge benötigen Sie zwei 40polige Frontstecker (Bestellnummer: 6ES7 392-1AM00-0AA0).
- Verdrahten Sie die Digitaleingänge 126.0 bis 126.3 wegen ihrer geringen Eingangsverzögerung immer mit geschirmten Leitungen.

**Vorsicht**

Bei Verdrahtungsfehlern an den Analogausgängen kann die integrierte Analogperipherie der CPU zerstört werden! (Z. B. bei versehentlichem Verdrahten der Alarmeingänge auf den Analogausgang.)

Der Analogausgang der CPU ist nur bis 15 V zulässig fest. Ausgang gegen Masse!

---

Hans Fonck  
Kaho St - Lieve  
TEI Kavala
4.4.3 Third problem

The third and last was the most important problem I had, and on which I searched many days. I made phone calls and e-mails to Siemens. Asked help to people I know in Belgium, here in Greece, everywhere where I thought I could find advice. There was a problem in the way I wrote the program.

I could run the program when I used only one line but when I used the two lines together, after the switch, the program blocked. The fact that there was no mention of an error made me conclude I was not far from the solution.

The pantheon was well disposed to me the next days. I can show you a program that works completely. The problem was that I created an inconsistency in the program. Like an eternal circle were you can not get out. So it blocked and the CPU did not know how to react.

The way of writing the program, as I had to deal with, is very important! Make sure that you don't give too many demands in one time.
The automation with Siemens SIMATIC S7

1998/99

4.5 The program:
Block: OBI organisationblock OBI

The 'call' for all the programs in the program.

Network: 1  FC1: general, Start / Stop

CALL FC 1

Network: 2  FC2: Set the parameters of the analog inputs.

CALL FC 2

Network: 3  FC3: Activate the alarms.

CALL FC 3

Network: 4  FC4: Activate line one or line two

CALL FC 4
FC2 - <Offline>

Name:  
Author:  
Time stamp Code:  
Interface:  
Length (Block / MC7 Code / Data):  

<table>
<thead>
<tr>
<th>Address Decl.</th>
<th>Name</th>
<th>Type</th>
<th>Initial Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>in</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>out</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>in_out</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>temp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

##### Block: FC2  Set parameters by means of a standard function of Siemens' A.I.

### Network: 1  Analog input HP

PIW 128 => signal PLC
70.0 bar = 10.0 V
67.0 bar = 0.0 V
reading out error in dataword MW200
HP between 67.0 and 70.0 bar => MD100

```
EN    "SCALE"  ENO
PIW128 -IN  RET_VAL- MW200
7.000000e+ 001 -HI_LIM  OUT- MD100
6.700000e+ 001 -LO_LIM
M1.0  BIPOLAR
```

### Network: 2  Analog input LP1

PIW 130 => signal PLC
20.5 bar = 10.0 V
17.5 bar = 0.0 V
reading out error in dataword MW202
LP1 between 17.5 and 20.5 bar => MD104

```
EN    "SCALE"  ENO
PIW130 -IN  RET_VAL- MW202
2.050000e+ 001 -HI_LIM  OUT- MD104
1.750000e+ 001 -LO_LIM
M1.0  BIPOLAR
```
Network: 3  Analog input LP2

PIW 130 => signal PLC
20.5 bar = 10.0 V
17.5 bar = 0.0 V
reading out error in dataword MW202
LP2 between 17.5 and 20.5 bar => MD108

<table>
<thead>
<tr>
<th>&quot;SCALE&quot;</th>
<th>EN</th>
<th>ENO</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIW132</td>
<td>IN</td>
<td>RET_VAL= MW204</td>
</tr>
<tr>
<td>2.050000e+001</td>
<td>-HI_LIM</td>
<td>OUT= MD108</td>
</tr>
<tr>
<td>1.750000e+001</td>
<td>-LO_LIM</td>
<td></td>
</tr>
<tr>
<td>M1.0</td>
<td>BIPOLAR</td>
<td></td>
</tr>
</tbody>
</table>
FC3 - <Offline>

Name: Author: Time stamp Code: Interface: Version: 00.01

Length (Block / MC7 Code / Data): 00222 00119 00000

<table>
<thead>
<tr>
<th>Address</th>
<th>Decl.</th>
<th>Name</th>
<th>Type</th>
<th>Initial Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>in</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>out</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>in_out</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>temp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Block: FC3  Activate the alarms

Network: 1  HP is to high
If the HP > 69.4bar (=8V) => alarm

<table>
<thead>
<tr>
<th>CMP &gt;=R</th>
<th>Q124.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD100 - IN1</td>
<td></td>
</tr>
<tr>
<td>6.940000e+</td>
<td></td>
</tr>
<tr>
<td>001 - IN2</td>
<td></td>
</tr>
</tbody>
</table>

Network: 2  HP is to low
If the HP < 68.2bar (=4V) => alarm

<table>
<thead>
<tr>
<th>CMP &lt;=R</th>
<th>Q124.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD100 - IN1</td>
<td></td>
</tr>
<tr>
<td>6.820000e+</td>
<td></td>
</tr>
<tr>
<td>001 - IN2</td>
<td></td>
</tr>
</tbody>
</table>

Network: 3  LP1 is to high
If the LP1 > 19.9bar (=8V) => alarm

<table>
<thead>
<tr>
<th>Q125.0</th>
<th>Q124.6</th>
<th>CMP &gt;=R</th>
<th>M10.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD104 - IN1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.990000e+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>001 - IN2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Network: 4  LP1 is to low
If the LP1 < 18.7 bar (=4V) => alarm

Q125.0  Q124.6  CMP <= R  M10.1
MD104  IN1
1.870000e+001  IN2

Network: 5  LP2 is to high
If the LP2 > 19.9 bar (=8V) => alarm

Q125.1  Q124.7  CMP >= R  M10.2
MD108  IN1
1.990000e+001  IN2

Network: 6  LP2 is to low
If the LP2 < 18.7 bar (=4V) => alarm

Q125.1  Q124.7  CMP <= R  M10.3
MD108  IN1
1.870000e+001  IN2

Network: 7  The LP1 or LP2 is to high
Set alarm Q124.2

M10.0  Q124.2
M10.2
<table>
<thead>
<tr>
<th>Network: 3</th>
<th>The LP1 or LP2 is too low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set alarm Q124.3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M10.1</th>
<th>Q124.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>M10.3</td>
<td></td>
</tr>
</tbody>
</table>
**Block: FC4 Activate line 1 or line 2 (=Bypass)**

If HP = OK => line 1 active, if not => de-activate
If EV1 HP completely open => led burning
If EV1 HP completely open and LP1 to low => alarm
If EV1 HP completely open and LP1 to high => alarm
If alarm LP is active => change to line 2 (=Bypass)
If the problem is solved => push reset and work again with line 1

**Network: 1**  
Open EV1 LP if HP is OK and line 1 is active

<table>
<thead>
<tr>
<th>Address</th>
<th>Decl.</th>
<th>Name</th>
<th>Type</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M30.0</td>
<td>Q124.0</td>
<td>Q124.1</td>
<td>M20.1</td>
<td>Q124.4</td>
</tr>
</tbody>
</table>

**Network: 2**  
Detection EV1 HP is completely open

| Q124.4  | I124.2 | Q125.0 |

**Network: 3**  
Open EV1 HP if HP is OK

| M30.0   | Q125.0 | M20.1 | M10.0 | M10.1 | Q124.6 |

**Network: 4**  
Close EV1 LP if LP is not OK

<table>
<thead>
<tr>
<th>M30.0</th>
<th>Q125.0</th>
<th>M20.1</th>
<th>M10.0</th>
<th>M20.1</th>
<th>M10.1</th>
<th>M20.2</th>
<th>S</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>R</th>
<th>S</th>
</tr>
</thead>
</table>
Network: 5  Open EV2 LF if HP is OK and line 2 is active

M30.0  Q124.0  Q124.1  M20.2  Q124.5

Network: 6  detection EV2 HP is completely open

Q124.5  I124.3  Q125.1

Network: 7  open EV2 HP if LP2 is ok

M30.0  Q125.1  M20.2  M10.2  M10.3  Q124.7
4.6 Remarks about the program:

The application I had to accomplish has space for many improvements in the cycle. It is only the basic I tried to program. For example implementing a device that reads the quantity of natural gas that passes through the pipe, or maybe use more valves that only work when there is a certain damage at the system. The use of a gas detector that controls if the system is safe to work with or not. That is necessary for the safety of the company and the working people.

There are of course a lot of improvements I did not mention because of my lack of knowledge about the theories of this subject. I’m certain specialists will give you a whole list of improvements and safety precautions that are required by the law at the present.

4.7 A word about the application of my project.

We find this kind of control system in many kind of areas in the industry. But finding the right way to explain or getting information about the subject is very difficult. The reason is very simple: each company has her own special specifications and they want to keep it this way!

The whole idea of my project is controlling the valves before and after the place where the pressure is reduced from high pressure to low pressure. In my case about 67 bar HP up to 19 bar LP. The drop is caused by a persistent between the two valves! When there is a problem in the current line we must have an switch to the bypass, so the problem can be solved by a mechanic. The bypass is needed because otherwise the system has to be stopped and that is a disadvantage for the factory.

Because I was not familiar with the use and the application of PLC’s this was a real challenge for me! I’m glad I took the job, it was not easy but that’s the way life goes! And not to forget, PLC has a great area with appliances in the industry.

My work is far from perfect and completed, it is just the basic way that is programmed. I do think lots of special qualifications can be added. Security valves, what if we have an overflow of natural gas,
But a big problem there is I think is that the pressure after the two LP-valves is not assured. We measure the pressure before the valves but not after. Maybe the use as only one, or as third LP-gauge can solve the problem.
When we use it as third it will be more as a control of the LP.
Think about this.

I hope the work I started is the beginning of a great project that can find its appliance in the industry.
Good luck to the person that takes the subject.
5. Conclusion.

Consideration and opinion about the project and the fact that it happened in Kavala!

An experience I will never forget!!!
I say not good bye but see you!
The first moments after my arrival I had a really difficult time here, the adjusting was very difficult. Not speaking the language, not knowing anybody and missing the people you love a lot.
But the ‘Erasmus-family’ helped me trough this period, and we became friends, let’s hope we never loose contact! Thanks.
By becoming more familiar with the habits of the local population I started to know many Greek people and they also became great friends. They made the life here very interesting and pleasant!
Thanks to them all.
At the end I had sometimes the impression that I lived here already for years. That gave me a strange feeling and made me wonder...

The project was a step in a for me unknown area and I enjoyed working with this kind of automation. In the beginning with the S7-212 PLC I made quick progress, what was good for the moral to work.
But with the S7-314IFM I had a lot of difficulties to make any progress. The reading of all the books about the possibilities, hardware, software and the program-language made me almost crazy.
I think you need a background to understand all the theory that it is written down in these books, so it is advisable to have more knowledge about programming and working with this Siemens-application.
But I have no regrets for choosing the subject. I liked it very much working with this kind of PLC. It was a whole new experience for me and I can say I learnt a lot!

The main fields of my studies don’t have the subjects of programming and working with PLC.
So this project is an expansion that I can add to my knowledge. I won’t say I can program very well, but I know the basic steps to make a meaningful program.

Thank you for given me this change of working with this important tool in the industry, I nevertheless enjoyed it the difficulties I had to overcome.

Hans Fonck
Kaho St.-Lieven
Biblio

- SIEMENS SIMATIC software:
  
  S7-300 Programmable controller:
  1. installation and hardware ⇒ manual
  2. user manual
  3. quick start
  4. ladder logic
  5. standard software
  6. program design

- internet: website with as subject: Siemens and natural gas
- Siemens Thessaloniki
- A few companies in Belgium.