THE AUTOMATION OF THE UNLOADING OR RECYCLING PROCESS OF THE FLY ASH FROM THE SILOS IN CEMENT MILLS

I. MĂRGINEANU[1] L.M. ITU[1]

Abstract: *Fly ash is one of the main ingredients used to produce some sorts of cement (like Portland cement). The fly ash manipulation system is made up of over 20 electric motors and 70 valves. In order to control these, only two function blocks will be created (one for the electric motors and one for the valves). In the unique program which controls the process, universal start/stop commands will be sent to the electric motors and open/close commands to the valves. These commands will be accepted if the corresponding values from an enabling vector are true. The particular program described in this article has been implemented for the fly ash storage system from Rupea used by the cement mill near Hoghiz.*

Key words: *cement silo, recycling, unloading.*

1. Introduction

In cement mills, silos are used to store cement or different ingredients which are then used to produce the cement. Fly ash is one of the main ingredients used to produce some sorts of cement (like Portland cement which is mostly used around the world). It is brought from thermoelectric power plants which run on coal. As fly ash is mostly available in winter and cement is produced during all seasons, there is the need to store the fly ash. The storing system is made up of a great number of silos, each one of them having a storing capacity of about 300.000 tons. For each of the silos there is an extraction valve and a system made up of channels and pipes which carry the fly ash to a buffer bunker. From here the fly ash is loaded into trucks in order to be brought to the location where it is mixed up with other ingredients which then make up the cement. In order to be transported through channels and then pipes, the fly ash has to be liquefied by blowing air into the silos, and the channels have to be fanned too. The air needed for these operations is provided by some compressors.

The channel system contains some valves which guide the fly ash to the pumps, which then carry the fly ash through pipes to the buffer bunker. The greater the number of silos is, the more complicated the channel system becomes and the greater the number of valves and fans has to be. Even though the number of identical parts (valves, fans, pumps, compressors, filters) is great, only few of those have to be active at the same time. Which of them are active depends on the silo which has been chosen to feed the buffer bunker.

This article presents the automation program for the unloading and recycling of

[1] Dept. of Automatics, *Transilvania* University of Braşov.

the fly ash and a method which determines which parts should become active using only one routine for the feeding of the buffer bunker and only one routine for the recycling process. Recycling is needed when the storage time exceeds a certain value and is done in order to avoid the fly ash to become compact in time.

2. Considerations on the Programming of the Application

The automation program is implemented on a Siemens PLC of the type SIMATIC S7 315-2DP, which is linked through a PROFIBUS network to distributed modules which collect data from the process, modules based on the module IM-153 [4]. The human-machine interface (HMI) is realized by an operating panel of the type MP 370 TOUCH which is connected to the PLC through the PROFIBUS network, too. The silos are grouped into four banks, each bank being made up of 12 silos. The channel system is much ore complicated than it is represented on the touch panel. The fly ash manipulation system is made up of very many actuators: over 20 electric motors and over 70 valves. A routine of the type FB [1], [3] (Function block) has been written (FB1 for the electric motors and FB2 for the valves) for each of those two classes and then a number of instances equal to the number of actuators have been created. The routine written for the electric motors (FB1) has the following inputs: a signal for availability (AV), a signal for working (B1), signals for local control (L1, LS - start and stop) and signals for program control (start - AUTO_RUN and stop - AUTO_STOP). There is an input which enables the controlling from the local panel of each electric motor (LP). The outputs of the routine control the contact maker which starts the electric motor (C1) and generate an alarm (M_ALARM).

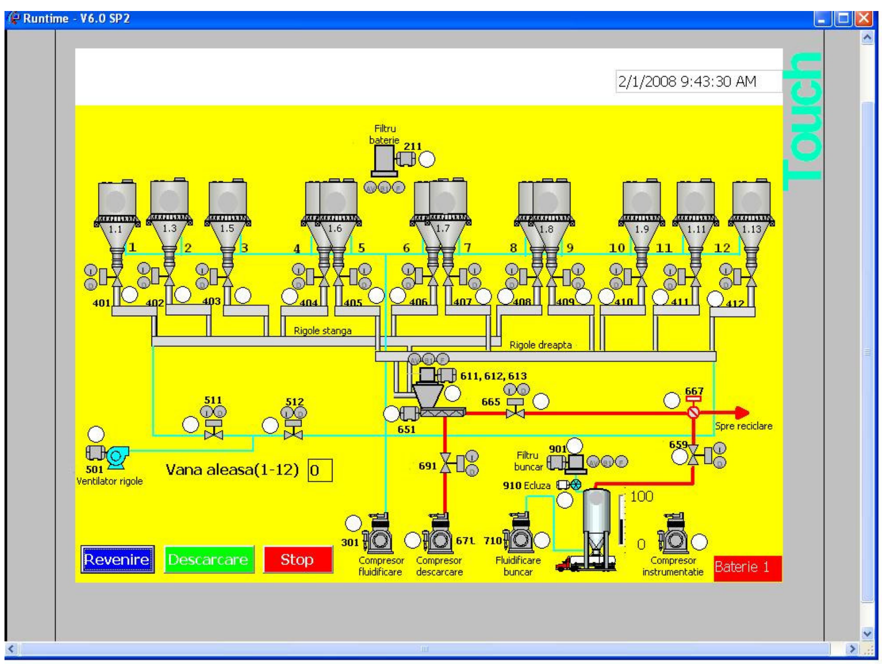

Fig. 1. *Human machine interface for bank*

The output M_ALARM becomes true if a start command has been sent to the electric motor and for the following two seconds the feedback signal B1 hasn't become true or if the e-stop was pushed. The routine written for the valves (FB1) has the following inputs: a signal for availability (AV), signals form the trip limits (ZSCx - for the closed state and ZSOx - for the opened state) and signals for program control (opening - OPEN_RQ and closing - CLOSE_RQ).

Each one of the routines mentioned above has an input which is connected to the signal from the e-stop (ESTOP) and which stops the actuator and generates an emergency signal in case the e-stop is pushed. The outputs of the procedure control the contact maker which opens the valve (OPEN) and generate the alarm signal (V_ALARM). The output V_ALARM becomes true if a command for opening or closing has been sent to the valve and the appropriate position hasn't been reached during the following 30 seconds.

Another routine, called FB3, has been used to fill the electric motor type objects on the HMI interface depending on their state. This routine provides a multiplexing variable which is used to fill these objects according to the LAFARGE standards (grey if the object isn't powered up; white if the object is available (AV); green if the object is running (B1) and flashing red if the object has suffered a failure (M_ALARM)). In order to fill the valves on the HMI interface the inputs and outputs of the program and the output of the FB2 routine called V_ALARM have been used as multiplexing variables.

The system provides two working regimes: manual and automatic. The two regimes can be chosen by means of the position of a button called AUTOMATICALLY/ MANUALLY which is placed on the start screen of the HMI interface (when the control system is started the default state is AUTOMATICALLY).

The installation is started by means of a button after the human operator has already chosen the silo from which the buffer bunker would be fed by entering the appropriate number into the corresponding input field. Based on this number, the program has to generate an enabling vector for all of the system's objects, vector which is made up of Boolean fields. A field in this vector becomes 1 (true) if the corresponding object takes part in the automation process when the chosen valve is opened and becomes 0 (false) if the corresponding object doesn't take part in the automation process when the chosen valve is opened.

The variable which determines the values of the vector is the identifier of the valve which will be opened. This is a number chosen by the human operator before the start of the unloading or recycling process by means of an input field which is placed on every screen of the interface. This number represents the input for a routine written in the SCL language, which provides the vector mentioned above. The routine is made up of a case instruction, which will determine different values for the fields of the enabling vector, values which ensue from the analyses of the technological process. The values from the enabling vector will start with the characters M_ for electric motors and V_ for valves, followed by the number of the electric motor or of the valve which is written down in the detailed scheme of the installation. In the unique program which controls the loading of the trucks or the recycling procedure, universal start/stop commands will be sent to the electric motors and open/close commands to the valves. The command is sent by the unique program and the enabling is done by means of the fields of the enabling vector.

For example, in order to generate the

pressure needed to circulate the fly ash through the channels it's necessary to start one of the two electric motors which train the two fans (501 or 502) and to open one of six valves (511 or 512 for bank 1 or 513 or 514 etc.) which guide the air to the channels. In the program, the routines which start the electric motors of the fans will have their enabling input connected to the left power bar through the enabling variables of the corresponding electric motors (Figure 2). In the same way, the routines which actuate the valves will have their enabling input connected to the left power bar through the enabling variables of the corresponding valves.

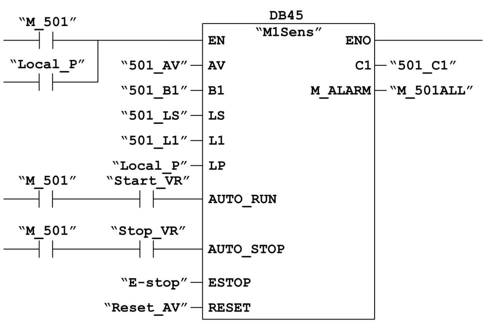

Fig. 2. *The call of the routine which controls the electric motor of the channel fan*

For safety reasons [2], the start/stop commands for the electric motors and the open/close commands for the valves will be sent by means of an AND function, one input of the function being the enabling variable. If the enabling variable isn't true, the routine won't be called in that cycle and no command will be sent to the actuators. The routines which control electric motor 501 and valve 511 are displayed in Figures 2 respectively 3.

On the start screen (having the switch AUTOMATICALLY/MANUALLLY on the position of AUTOMATICALLY) first you press the button called BATTERY *number* from the unloading group. It's the number of the bank from which you want to do the unloading.

If the number of the bank is 1, the screen displayed in Figure 1 will appear which allows you to control the unloading process from the silos 1.1-1.13. First you chose the valve which should be opened by pressing the input field called *chosen valve*. When pressing firmly this field, a keyboard will appear which will allow you to enter the desired number of the valve. Afterwards you may press ENTER to continue. You can only chose valves which are available for the selected bank of silos, otherwise an error message will appear and the typed number won't be accepted. If the valve has been chosen correctly you may press the button called UNLOADING**.** This would lead to the following actions:

1. The unloading process starts for the silo corresponding to the chosen valve.
2. The buttons UNLOADING and RETURN and the input field of the screen will become invisible, so that it's impossible to chose another valve if the unloading process has already started. The objects mentioned above will become visible again only when the unloading process is finished. The installation starts only if the buffer bunker is not full. If this condition is fulfilled, an alarm horn will toot for 15 seconds before the installation starts.
3. The equipments which take part in the unloading process have to be started in a certain order:

a) The instrumentation compressor is started;

b) The liquefaction compressor is started (Start_CF: 301 or 302);

c) The air pass for the channels is opened (Desc_VV: 511, 512 … 518);

d) The fan for the channels is started (Start_VR: 501, 502);

e) The input valve of the fly ash pump is opened (Desc_VAP: 691, 692, 693, 694);

f) The output valves of the fly ash, which lead to the buffer bunker are opened (Desc_VIbau: 665, 664, 663, 662);

g) The valves which connect the fly ash pump and the buffer bunker are opened (Desc_VBA: 659, 658);

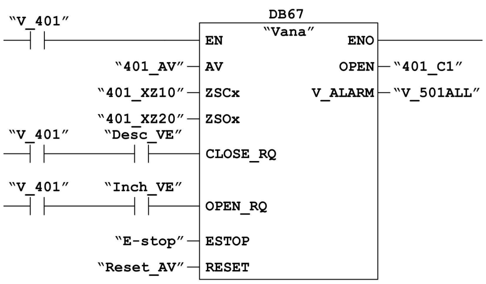

Fig. 3. *The call of the routine which controls valve 401*

h) The unloading compressor is started (Start_671);

i) The filter and the programmer of the fly ash pump are started (Start_MFI. The programmers are started corresponding to the enabling vector after the start of the motor of the pump and only if they are available. This is done in the block OB1);

j) The filter and the programmer of the buffer bunker are started (Start_901. The programmer is started in the block OB1 after the motor of the filter has already been started and only if the programmer is available);

k) The gate of the filter from the buffer bunker is started;

l) The extraction valve is opened.

4. The process continues until one of the following events occurs:

a) The buffer bunker is full (the information will be obtained from a level transducer);

b) The pressure in the fly ash pump exceeds the allowable limit (information obtained from the pressure transducer of the pump);

c) The stop button for the unloading process has been pressed on the HMI interface (the button Stop);

d) An emergency occurs for one of the actuators.

5. If the process has stopped because of a signal from the level transducer of the buffer bunker, the following actions will be executed:

a) The extraction valve is opened;

b) After a period of time, which is determined experimentally, the unloading compressor is stopped;

c) The connection valves between the fly ash pump and the buffer bunker are closed;

d) The air valve of the fly ash pump is closed;

e) After a period of time, which is determined experimentally, the fan for the channels is stopped;

f) The air pass for the channels is closed;

g) The system remains now in a wait state which can be left if the buffer bunker is voided or if the stop is pressed. In the first case the program starts over again from step 3c. If the stop button is pressed, the filter and the programmer of the fly ash pump, the filter and the programmer of the buffer bunker and the two compressors (the instrumentation compressor and the liquefaction compressor) are stopped. The program must rejoin the initial state, waiting for a new start.

6. If the pressure in the fly ash pump exceeds the allowable limit, the extraction valve is closed and the system waits until the pressure drops under a certain limit and then the program restarts with step 3l.

7. If the stop button is pressed, all the parts of the system are stopped in reverse order.

8. If an emergency has occurred for one of the active parts, the program will rejoin the initial state and the system will act according to the emergency routines.

In order to avoid the opening and closing of the extraction valves being done with a high frequency, the pressure for the pump is tested using the SCHMIDT routine, which is an implementation with two comparators of a trigger SCHMIDT. The inputs of the routine are: the controlled pressure, which is obtained from analogical input modules and the two threshold values: ON_TO_OFF and OFF_TO_ON where the state of the Boolean output changes (Figure 4). If the characteristic of the pressure transducer is known, the values of the input variables can be specified directly in converter units, being no need to convert them into the international system.

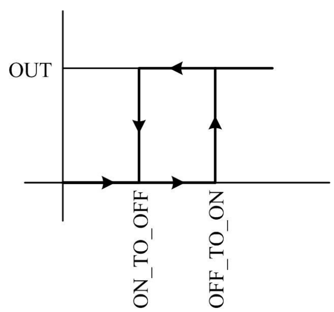

Fig. 4. *Trigger SCHMIDT*

3. Conclusions

The described method may be applied for all sequential automations with great numbers of identical parts, which take part in similar actions, but at different times. This means you have to define an object for every type of part, using the language of the particular PLC. For the automation described in this work, the actuators are the electric three-fazed alternative current motors and the pneumatic valves; the objects have been implemented by means of procedures of the type FB in Step 7 Manager.

The original contribution of this paper is that all actuators which take part simultaneously in the automation process are controlled by a unique algorithm. This allowed a substantial reduction of used memory and offers the possibility of extending automation just by changing validation vector.

All of the actuators, which are grouped into categories, take part in this algorithm but only one group is activated at a time. The call of the procedure or a certain part is managed by the fields of an enabling vector. The algorithm which computes the fields of the enabling vector depends on the specific structure of the installation. This type of design for automation is particularly flexible; it diminishes the dimension of the code and enables future developments of the automation program with less effort. The particular program described in this article has been implemented for the fly ash storage system from Rupea used by the cement mill near Hoghiz.

References

1. Mărgineanu, I.: *Programmable Logic Controllers*. Cluj-Napoca. Editura Albastră, 2005.
2. Phipps, C.: *Fundamentals of Electric Control*. TWI Press, Inc., 2000.
3. Stereson, J.: *Fundamentals of Programmable Logic Controllers. Sensors and Communication*. 4th edition, Pearson Education, 2002.
4. *** *System Software for S7-300/400. System and Standard Function. Reference Manual*. Edition 03/2006.