Hierarchical Control of a Complex Conveyors Belt System

CORINA MARIA DINIŞ GABRIEL NICOLAE POPA ANGELA IAGĂR
Department of Electrotechnical Engineering and Industrial Informatics
Politehnica University Timişoara
Revoluției Str., no5, Hunedoara, 331128
ROMANIA
corina.dinis@fih.upt.ro, http://www.fih.upt.ro

Abstract: - Using the software-oriented technologies which contain maximum of integrated software resources in product, in this work has been developed the software for the hierarchical control of a complex conveyors system. This ensures hardware simpleness, implementing of some evolved control algorithms, on-screen measuring devices, a friendly graphic interface, which makes that the entire system to show high reliability and flexibility. It shows the advantages of microcontroller installations comparatively with classical control installations.

Key-Words: - Software-oriented hierarchical control, belt conveyors system, microcontroller

1 Introduction

The continuous transportation process using belt conveyors is one of the most used solutions in various activity fields, starting with sorting of substances from industrial shops, going further with material processing from the sintering plants and ending with flexible robotized lines or the conveyors flow from airports.

As overview, the control of such transportation flow is very complex, both concerning the necessary hardware equipment and the implementation of control algorithms [1,3,7,9].

Nowadays, the major part of conveyors' control systems are hardware-oriented. These contain many hardware components, such as: electromechanical elements, logical and analogical devices, discrete components, power electronics, etc. The control program is reduced to minimum and is implemented, usualy, on the microcontroller [4].

A microcontroller is an integrated circuit that achieves many of a typical computing system's functions. This contains in a single chip a microprocessor as central processing unit, memories, counters, converters, input-output peripherals. One of the advantages of the systems with microcontroller is using the software to replace a complex logic, without modifying anything in the hard structure. Another advantage is that it uses software instead of some complex and expensive hardware components. Thus, by using microcontrollers are saved the costs due to some expensive hardware components, and the necessary physical space is much more reduced.

In many cases, the hardware equipment reduces the reliability and flexibility of the entire system. On the other hand, this control system, together with the power electronics, is higher in volume and price than the working machine itself.

Development of computerized technologies emphasized that many control elements, controls, automation, etc. of mechanical, electrical and electronical nature can be achived by program, reducing the complexity and saving materials and labor, with positive effect on cost and safety in operation [8].

Thus has been developed a new trend, which is currently dominant in designing of technical stystems, called „software-oriented technology". This technology contains maximum of integrated software resources in product (software-embedded) and minimum of hardware resources, the software resources being in fact products of human intelligence, materialized in programs and included in the equipment, forming its artificial intelligence [3,5,6]. In this work is presenting a software-oriented control system, **hierarchycally distributed** and **conceived** for a belt conveyors flow, with application for material dosing.

2 Control of a belt conveyors system, with MC68HC05B6 - Motorola microcontroller

Microcontroller MC68HC05B6 (Motorola) is an HCMOS integrated circuit, that has the following resourses:

- 8-bit arithmetic and logical unit, that can execute adding, subtracting, multiplying, incrementing, decrementing, logical and rotating operations;
- 176 RAM 8-bit;
- 8 kbytes EEPROM for programming;
- 256 EEPROM 8-bit;
- 16 bi-directional inputs/outputs;
- counter of 16 bits;
- 2 capture inputs;
- software reset of the main counter;
- 8 channels for 8-bit analogue-digital conversion;

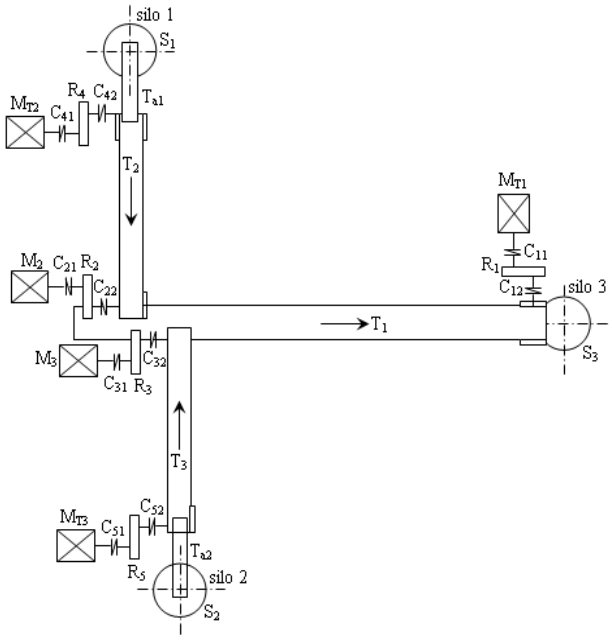

Fig.1. Branched system of conveyor belts

- 2 conversion channels 8-bit digital-analogue;
- serial asynchronous communication interface.

The conveyor belts T_2 and T_3 are supplied with material from silos S_1 and S_2. These are discharging the charge on the basic belt T_1. The transport flows on the belt system are on directions T_{a1}-T_2-T_1, respectively T_{a2}-T_3-T_1 indicated by arrows.

In fig.2 is presenting the power and control diagram of the conveyor belt system.

The power part contains five motors (M_{T1}, M_{T2}, M_{T3}, M_{Ta1}, M_{Ta2}) for driving each conveyor belt and belt feeders (fig.1). Each motor is protected by fusible fuses (F_{12}, F_{22}, F_{32}, F_{42}, F_{52} against short-circuits) and by thermo-bimetallic relays (F_1, F_2, F_3, F_{a1}, F_{a2} for overload protection). As commutation elements are used contactors (K_1, K_2, K_3, K_{a1}, K_{a2}). The contactors' coils are supplied at 380 V c.a., being controlled by the normally open contacts of the microrelays K_{11}, K_{21}, K_{31}, K_{a11}, K_{a21} by the system with microcontroller (fig.2). Microrelays K_{11}, K_{21}, K_{31}, K_{a11}, K_{a21} are controlled by 1 logic. The system by microntroller is supplied at 5V. All components were connected at the first 16 digital ports (P_1,…,P_{16}). The normally closed contacts of the thermal protections (F_1, F_2, F_3, F_{a1}, F_{a2}) were connected by resistances of 10 kΩ to 5V. Thus, if the protections don't act, their normally closed contacts do not open and the microcontroller interprets this status as 0 logic. If a protection is actuating, the related contact is open and on the respective input is applied signal 1 logic.

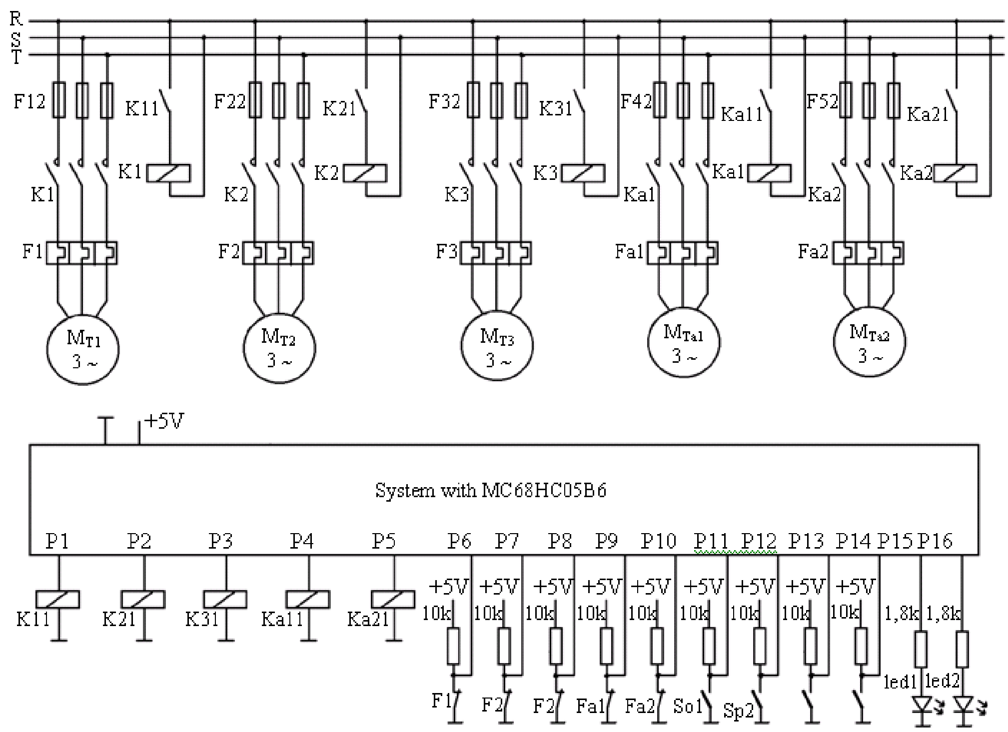

Fig.2. The power and control diagram of the conveyor belt system

The two flows can be controlled separately: flow 1 (T_1-T_2-T_{a1}) and flow 2 (T_1-T_3-T_{a2}). Flow 1 is started by button S_{p1}, and flow 2 by button S_{p2}. Flows' stopping is made by the stop buttons S_{o1}, respectively S_{o2}. If the buttons are not actuated, it's interpreted 1 logic, and if they are closed, is interpreted 0 logic. There are also connected two LEDs – LED1 and LED 2- that indicates the operation of flow 1 respectively flow 2. The LEDs' control is made by 1 logic.

In fig.3 is presenting the main control & command program of the conveyor belt system.

There are deactivated (put on 0 logic) all the digital ports (P_1,…,P_{16}). Then, is checked if S_{p1} (start buton of flow 1) is closed. If yes, the LED 1 turns-on and is passing to the flow1 subroutine. Otherwise is checked if S_{o1} (stop button for flow 1) is closed. If yes, LED 1 turns-off, the contactor K_{a1} switches-out (feeder T_{a1} stops), then follows a break of 0,5s, the contactor K_2 switches-out (conveyor T_2 stops), then follows a break of 0,5s and contactor K_1 switches-out (conveyor T_1 stops). In this way are stopped delayed the conveyors from flow 1.

If S_{o1} is open, then are checked the thermal protections of motors from flow 1. If the protection of any motor from flow 1 (F_1 or F_2 or F_{a1}) is actuated, then are switched-out undelayed all the conveyor belts (either from flow 1 and from flow 2) and is skipping to label 10.

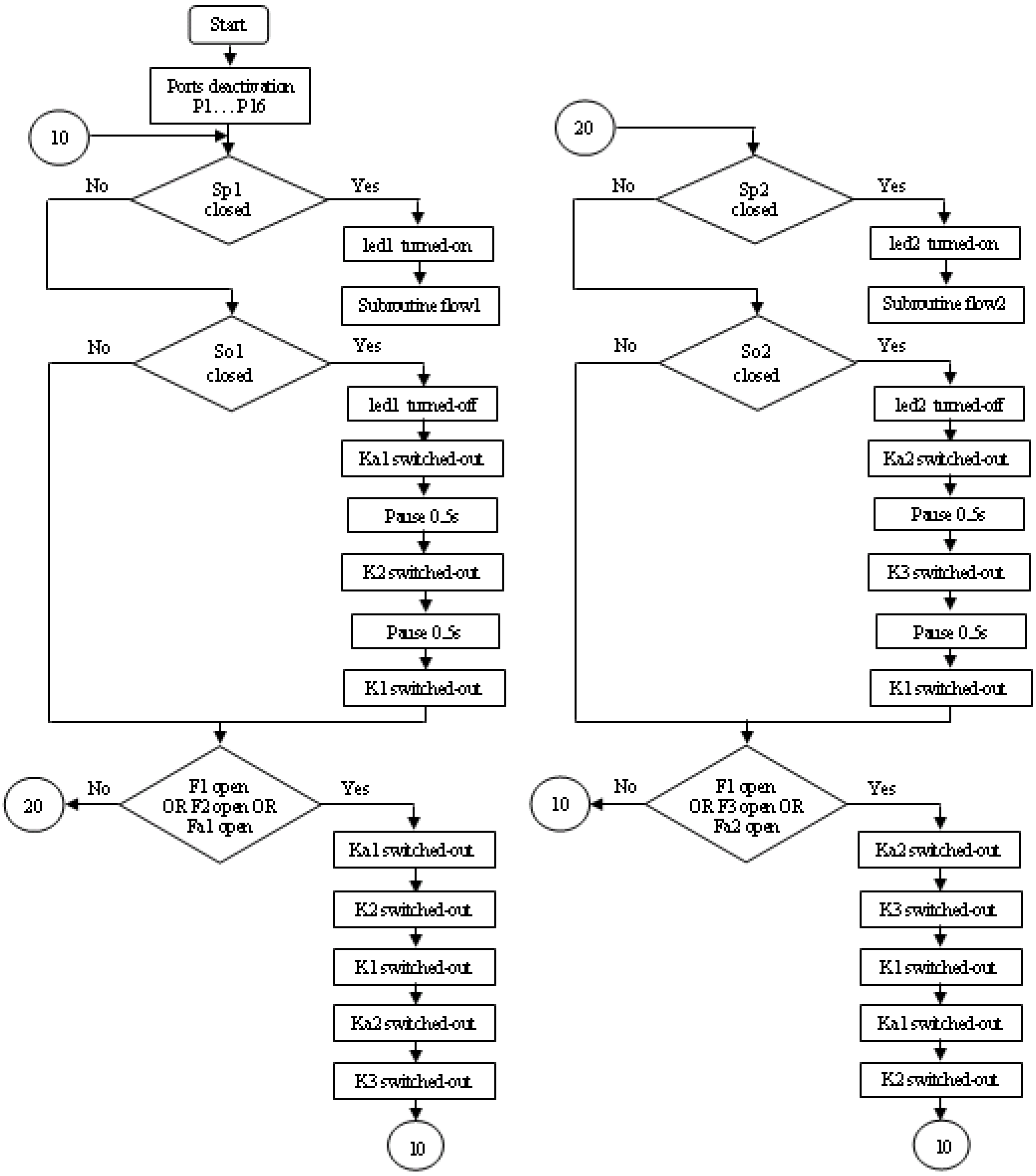

Fig.3. Main program

Thus, is checked if S_{p2} (start button of flow 2) is closed. If yes, LED 2 turns-on and is passing to flow2 subroutine. In this way, is checked if S_{o2} (stop button for flow 2) is closed. If yes, LED 2 turns-off, the contactor K_{a2} (switches-out feeder T_{a2} stops), then follows a break of 0,5s, the contactor K_3 switches-out (feeder T_3 stops), then follows a break of 0,5s after which the contactor K_1 switches-out (feeder T_1 stops). In this way are stopped delayed the conveyors from flow 2. If S_{o2} is open, then are checked the thermal protections of motors from flow 2. If the protection of any motor from flow 2 (F_1 or F_3 or F_{a2}) is actuated, then are switched-out undelayed all the conveyor belts (either from flow 1 and from flow 2) and is skipping to label 10.In fig.4 is presented the flow1 subroutine. K_1 switches-in (T_1 starts), follows break of 2s. If F_1 (thermal protection) is open, K_1 switches-out (T_1 stops), LED 1 turns-off and is skipping to label 10. If F_1 is closed, K_2 switches-in (T_2 starts), it follows a break of 2s. If F_2 (thermal protection) is open, K_2 switches-out (T_2 stops), K_1 switches-out (T_1 stops), LED 1 turns-off and is skipping to label 10. If F_2 is closed, K_{a1} switches-in (T_{a1} starts), it follows a break of 2s. If F_{a1} (thermal protection) is open, K_{a1} switches-out (T_{a1} stops), K_2 switches-out (T_2 stops), K_1 switches-out (T_1 stops), LED 1 turns-off. Is reverting from subroutine.

In fig.5 is given the logic diagram of flow2 subroutine, which in principle is the same as the one of flow1 subroutine.

Is presented the control program's listing of the conveyor belt system achieved in CCBasic.

```
‘Program for controlling the conveyor belts
define ports wordport[1] 'Ports 1 - 16
define t1 port[1] 'Port command contactor K1, for
conveyor T1
define t2 port[2] 'Port command contactor K2, for
conveyor T2
define t3 port[3] 'Port command contactor K3, for
conveyor T3
define ta1 port[4] 'Port command contactor Ka1, for
feeder Ta1
define ta2 port[5] 'Port command contactor Ka2, for
feeder Ta2
define f1 port[6] 'Port thermal protection motor
conveyor T1
define f2 port[7] 'Port thermal protection motor
conveyor T2
define f3 port[8] 'Port thermal protection motor
conveyor T3
define fa1 port[9] 'Port thermal protection motor
feeder Ta1
define fa2 port[10] 'Port thermal protection motor
feeder Ta2
define sp1 port[11] 'start-up button flow 1
define so1 port[12] 'shutt-down button flow 1
define sp2 port[13] 'start-up button flow 2
define so2 port[14] 'shutt-down button flow 2
define led1 port[15] 'LED operation flow 1
define led2 port[16] 'LED operation flow 2
'Main Program
'Initializations
    ports = off
10  if sp1=off then led1=on
    if sp1=off then gosub flow1
    if so1=off then led1=off
    if so1=off then ta1=off
    pause 25
    if so1=off then t2=off
    pause 25
    if so1=off then t1=off
    if f1=on or f2=on or fa1=on then ta1=off
    if f1=on or f2=on or fa1=on then t2=off
    if f1=on or f2=on or fa1=on then t1=off
    if f1=on or f2=on or fa1=on then ta2=off
    if f1=on or f2=on or fa1=on then t3=off
    if sp2=off then led2=on
    if sp2=off then gosub flow2
    if so2=off then led2=off
    if so2=off then ta2=off
    pause 25
    if so2=off then t3=off
    pause 25
    if so2=off then t1=off
    if f1=on or f3=on or fa2=on then ta2=off
    if f1=on or f3=on or fa2=on then t3=off
    if f1=on or f3=on or fa2=on then t1=off
    if f1=on or f3=on or fa2=on then ta1=off
    if f1=on or f3=on or fa2=on then t2=off
    goto 10
    ‘Subroutine flow1
    #flow1
    t1=on
    pause 100
    if f1=on then t1=off
    if f1=on then led1=off
    if f1=on then goto 10
    t2=on
    pause 100
    if f2=on then t2=off
    if f2=on then t1=off
    if f2=on then led1=off
    if f2=on then goto 10
    ta1=on
    pause 100
    if fa1=on then ta1=off
    if fa1=on then t2=off
    if fa1=on then t1=off
    if fa1=on then led1=off
    return
    ‘Subroutine flow2
    #flow2
    t1=on
```

```
pause 100
if f1=on then t1=off
if f1=on then led2=off
if f1=on then goto 10
t3=on
pause 100
if f3=on then t3=off
if f3=on then t1=off
if f3=on then led2=off
if f3=on then goto 10
ta2=on
pause 100
if fa2=on then ta2=off
if fa2=on then t3=off
if fa2=on then t1=off
if fa2=on then led2=off
return
```

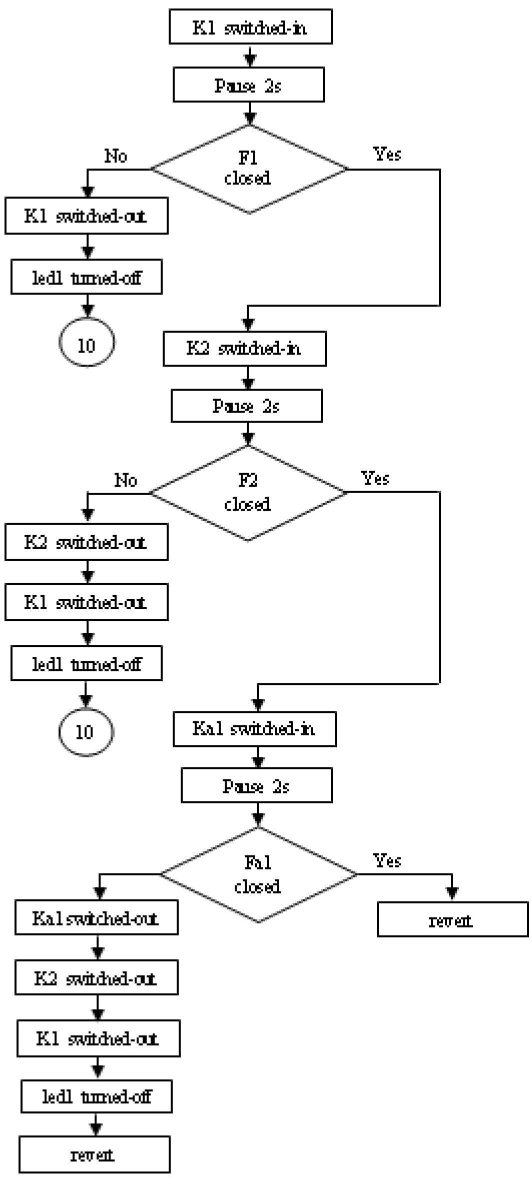

Fig.4. Flow1 subroutine

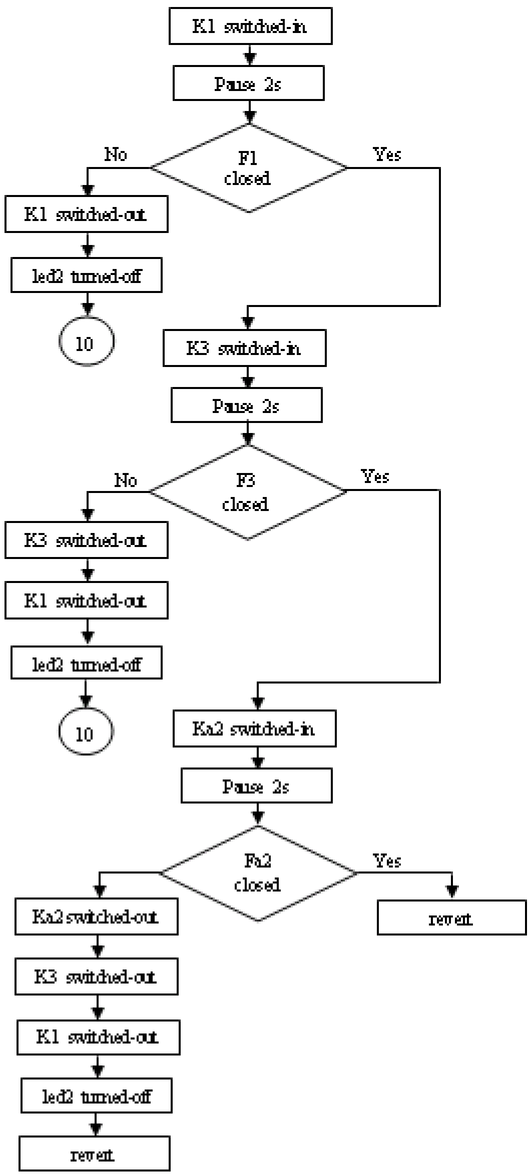

Fig.5. Flow2 subroutine

3 Hierarchical Control of a Belt Conveyors System

In general case, a conveyor system contains the following main elements: a bus-belt, one or more collector belts, and two or more working belts. On the two working belts are delivered: the hot return from the return bunker (BRET) and the limestone, coke and iron ores from BKCM bunker. The entire charge mixture is overflown from the bus-belt into the charge bunker (BSRJ). The presence of these belts is controlled by level transducers. Each conveyor is served by three transducers: of speed, overflow and emergency and is driven by a motor, being present also a preventive signalling horn.

To achieve the control by software orientation method, will be used a hierarchical structure on three levels, having at the superior level an industrial PC, at the mid level a number of PLCs, one for each conveyor, a data execution and feedback level the working machines and the process transducers.

The programable logical controls (PLC_i) are developped based on Keil MCB167-NET kit, having the following characteristics: 5 I/O ports, 1 input port, two serial communication channels (ASC0 asynchronous/synchronous serial channel and a high speed synchronous serial channel SSC), 1 x RS232 Interface, 2xCAN Interfaces, 1x Ethernet Controller.

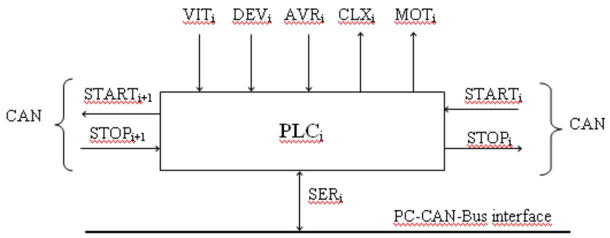

Fig.6 Block-diagram PLC_i

The local display of the conveyor's condition is achieved by a liquid chrystal display (LCD).

In fig.6 is shown the block-diagram of the PLC_i afferent to the “i” conveyor and the elements which it serves [2].

For a T_i conveyor have been made the following notes:

VIT_i – speed transducer;
DEV_i – overflow transducer (filling-up);
AVR_i – emergency transducer (in-line);
SER_i – serial communication between PLC and PC;
$START_i$ – start signal for PLC_i;
$STOP_i$ – stop signal for PLC_i;
$START_{i+1}$ – start signal for the upstream conveyor;
$STOP_{i+1}$ – stop signal for the downstream conveyor;
CLX_i – control signal for the horn;
MOT_i – control signal for starting the motor or motors group.

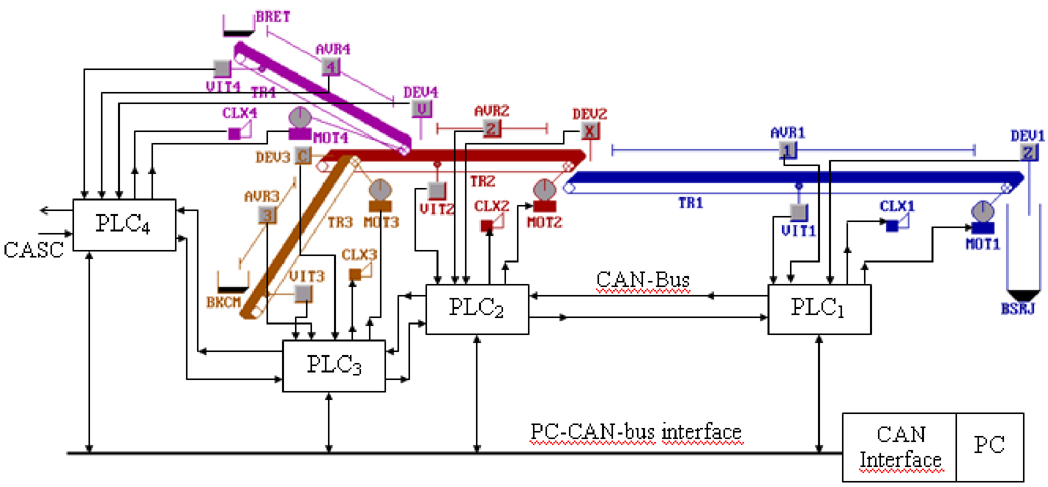

Fig.7 Structure of the hierarchical control system

The control algorithm implements the following 14 conditions:

- Completely automatic operation with accoustic and graphic signalling in real-time;
- At commissioning, the correctness of its integrity is tested;
- Starting-up is made on the routes depending on the material's presence transducer;
- Starting-up of a conveyor is made only after it was made the preventive accoustic signalling;
- The minimum value of the material from the bunker stops the route in the direct sense of the material flow;
- The duration of a conveyor's starting-up is controlled by the speed transducer through the Watchdog;
- The upstream conveyor starts-up after the one from the downstream has started;
- If a conveyor has stopped, all the others which overflow on it will stop, too;
- The upstream conveyor will be stopped if the downstream bunker has filled-up;

The conveyor will stop if the emergency transducer AVR_i is actuated;

- If the Watchdog has been actuated, the conveyor will stop;
- Emergency detection produces the locking-up of starting the belt until the remedy of the defect;
- Signalling of the system's condition is ensured at superior and local level;
- Signalling of the transducers condition is ensured at superior and local level.

Structure of the hierarchical system is presented in fig.7 and can be extended for any configuration by cascade connections (CASC) [1].

4 Modeling and Implementation of the Hierarchical Control Software

Testing of the control system's feasability was done at the beginning by modeling and simulation of PLCi in the MatLab-Simulink. In this respect, was developped the mathematic model which was used as basis for development of Simulink model from fig.8.a. In fig.8.b. are shown the simulation results of an individual PLC. Based on the model and the simulation results, it started the development and implementation of the control software. The control software at high level was achieved in I80X86 assembling language and contains about 8000 instruction lines. The program ensures a real-time graphic interface on PC's screen from the dispatcher, with the following elements [2]:

- Representing of the transportation flow in real time;
- Real condition of the transducers;
- The condition of each belt;
- Automatic or manual control;
- Chosing a route in automatic mode, depending on the material from bunkers.

Further are shown few program sequences, i.e. the data segment and the main control routine.

```
.model small
.stack
.data
titleApp          db   " Control a conveyors system",0
titleKeys         db   " Keys used:",0
titleMessages     db   " Messages:",0
titleState        db   " State transducers:",0
trsp1             db   " TR1",0
trsp2             db   " TR2",0
trsp3             db   "  TR3",0
trsp4             db   "  TR4",0
damage            db   " AVR:",0
discharge         db   " DEV:",0
speed             db   " VIT:",0
MesDamage         db   " DAMAGE",0
MesSintering      db   " SINTERING",0
MesSpeeding       db   " SPEEDING",0
MesSpeedNormal    db   " NORMAL SPEED",0
MesSpeedAbnormal  db   "ABNORMAL SPEED",0
MesRest           db   " REST",0
MesBNCfull        db   " FULL",0
MesBNCempty       db   " EMPTY",0
MesBNCprocessed   db   " MEDIUM",0
. code
; main program
start:   call portzero          ; initial state
         call getcar
         call getcar8x16
         mov ax,@data
         mov ds,ax
         call far ptr whichvga
         call svgamode
         call doInitScreen      ;virtual instruments
         call MainLoop          ; monitoring
exitDOS: call portzero          ; initial state
         call txtmode
         mov ax,4c00h
         int 21h
         end start
```

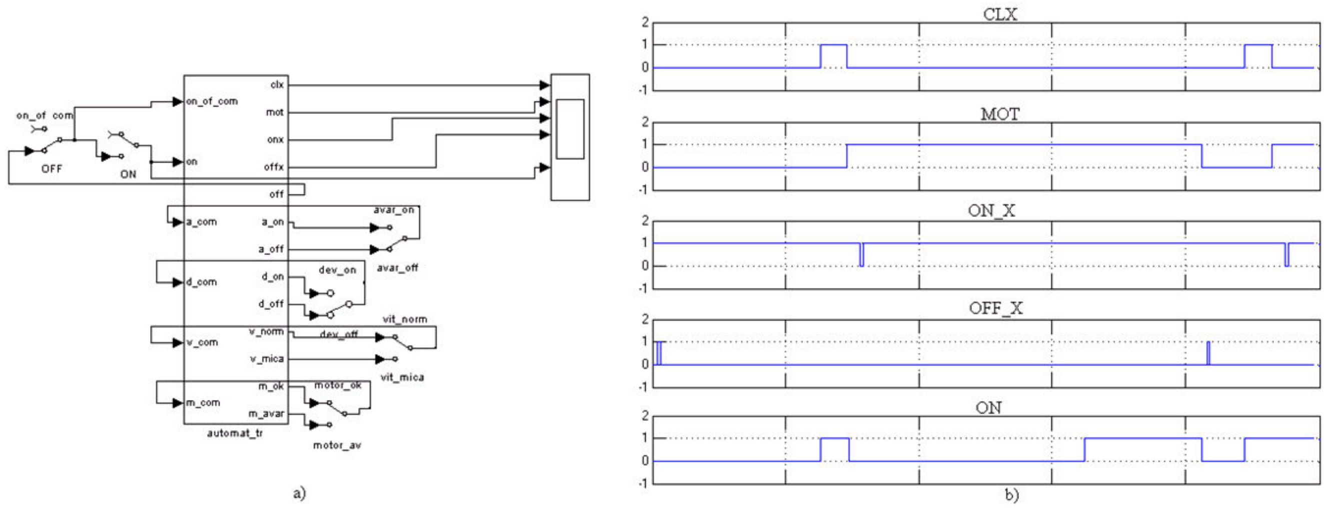

Fig.8 Simulink Model for PLC_i

In fig.9 are shown the results of modeling the conveying flow in MatLab-Simulink.

One can see the variation in time of the control and feedback signals, in accordance with the imposed technologycal conditions.

In fig.10, 11, 12, 13 is shown the work screen of the PC from the superior level in start regime for a configuration of 4 conveyors from a sintering plant.

When the materials' presence transducer detects the superior level in the BRET return bunker or in the mixing bunker of limestone, coke and iron ores BKCM, the conveyors system starts as follows:

- PLC_1 gives the preventive accoustic warning command to the CLX1 horn, then is given the start command for MOT1 motor, which drives the conveyor 1 (TR1);
- The speed transducer VIT1 detects the normal operation regime of the TR1 conveyor, and PLC1 will give the START command for PLC2;
- PLC2 controls the preventive accoustic warning by CLX2 horn, then is given the start command for MOT2 motor of the conveyor 2 (TR2);
- The speed transducer VIT2 detects the normal operation regime of the TR2 conveyor, and PLC2 will give the START command for PLC3 and the process continue;
- Any emergency detected by the transducers generates the initiation of STOP subroutines, by displaying the condition on the PC screen.

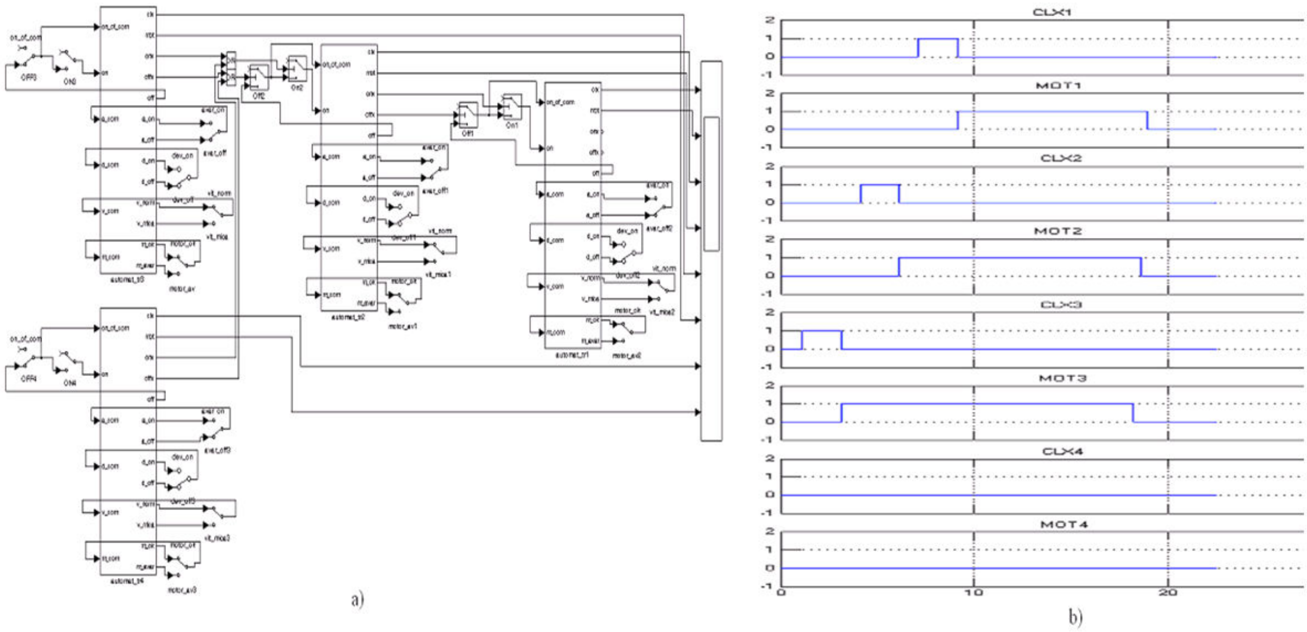

Fig.9 Simulink model for simulations

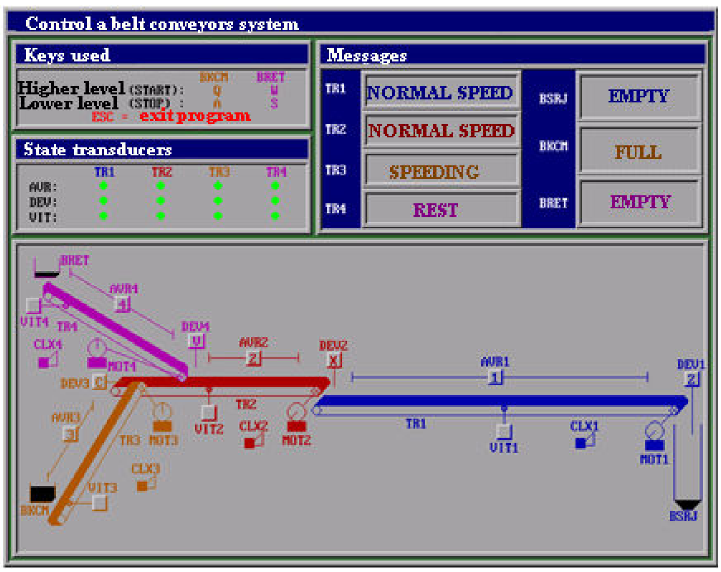

Fig.10 Work screen when conveyors 1 and 2 are started

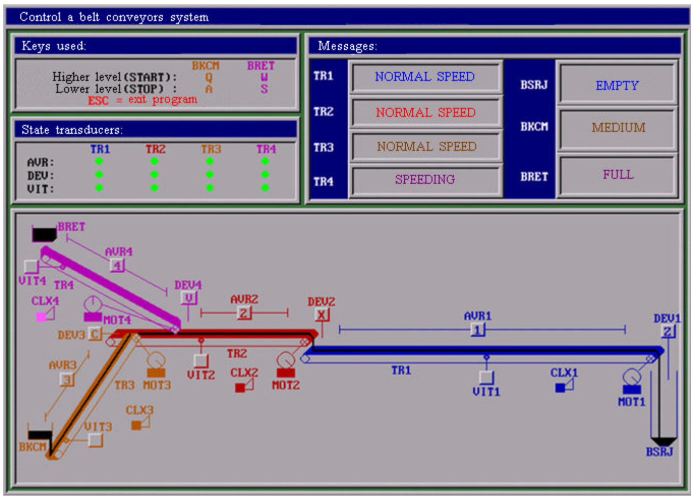

Fig.11 Work screen when conveyors 1, 2 and 3 are started and discharged from BKCM bunker

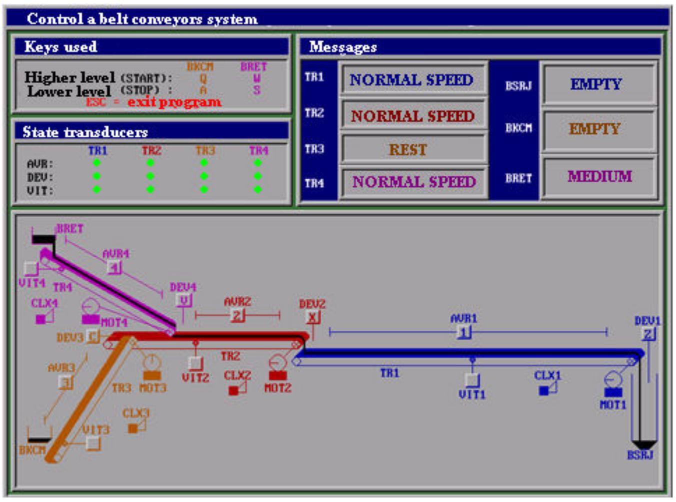

Fig.12 Work screen when transporters 1, 2 and 4 are started and discharged from BRET bunker

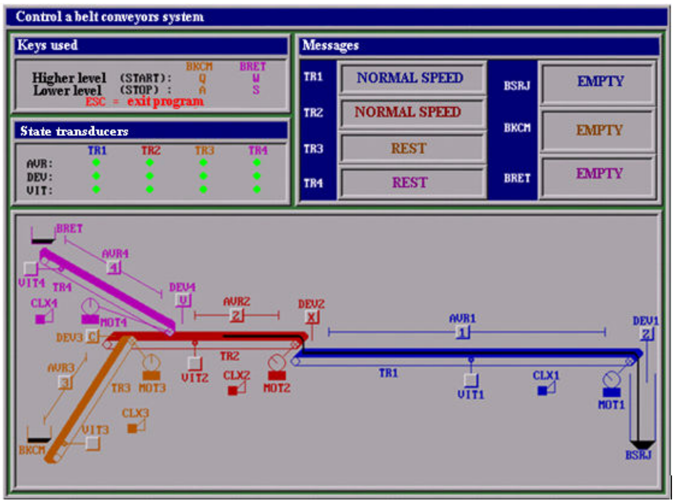

Fig.13 Work screen before the shutting-down of the belt conveyor system

5 Conclusion

Without the control systems by contacts and relays or by logical gates, the control by microcontroller ensures flexibility in programs' achievement and is achieved by programming without intervene upon the hard components.

In this work was conceived and developped a flexible control system of the conveying flow, based

on software-oriented technology. Has been achieved a hierarchical control structure of the conveying flow. It was modeled and simulated in MatLab-Simulink with good results. Based on the model, was conceived a control software of the belt conveying process, adaptable to any configuration in space of the conveyors.
The results of this work may be a support for revamping the belt conveying flows used in the sintering plants, lignite quarries or for the distribution conveyors from the robotic manufacturing cells, etc.

References:

[1] C.M. Diniş, E. Pop, A. Iagăr, G.N. Popa, Researches upon the hierarchical control of a conveyor belt system from the sintering plants, *Metalurgia International*, Vol. XIII, No.9, 2008, pp. 54-61.

[2] C.M. Diniş, *Researches concerning modelling and management of the processes from sintering plants*, Doctorate Thesis, University of Petroşani, The Faculty of Mechanical and Electrical Engineering, Romania, 2004.

[3] J.K.L. Ho, P.G. Ranky, Object oriented modelling and design of reconfigurable conveyors in flexible assembly systems, *International Journal of Computer Integrated Manufacturing*, Vol. 10, Issue 5, September 1997, pp. 360-379.

[4] A.L. Antipov, N.A. Malov, Model of an adaptive hierarchical control system, *Proceedings of SPIE*, Volume 5851 Fundamental Problems of Optoelectronics and Microelectronics II, June 2005, pp. 209-212.

[5] E. Zamai, A. Chaillet-Subias, M. Combacau, An architecture for control and monitoring of discrete events systems, *Computers in Industry*, Volume 36, Issues 1-2, 30 April 1998, pp. 95-100.

[6] M.R. Da Silveira, M. Combacau, Supervision and control of heterarchical discrete event systems, *Controle & Automação*, Vol. 17, no. 1, Janeiro, Fevereiro e Março 2006, pp. 1-9.

[7] J.O. Meré, F.A. Elías, A.M. Gonzáles, M.C. Limas, F.J.M.P. Ascacíbar, Data mining and simulation processes as useful tools for industrial processes, *Proceedings of the 5th WSEAS Int. Conf. on Simulation, Modeling and Optimization,* Corfu, Greece, August 17-19, 2005, pp. 243-249.

[8] C. Lupu, D. Popescu, A Udrea, Real-time control application for nonlinear processes based on adaptive control and the static characteristic, *WSEAS Transaction on Systems and Control*, Issues 6, Vol. 3, June 2008, pp. 607-616.

[9] J. Aguilar, M. Cerrada, F. Hidrobo, J. Chacal, C. Bravo, Specification of a multiagent system for planning and management of the production factors for automation based on the SCDIA framework and MASINA methodology, *WSEAS Transactions on Systems and Control,*Issues 2, Vol. 3, February 2008, pp. 79-88.