PLC-BASED HOLONIC MANUFACTURING CELL TRANSPORT SYSTEM

Andrei ROŞU[1]

Această lucrare descrie o abordare ştiinţifică pentru controlul unui sistem de transport integrat într-o celulă de fabricaţie utilizând un singur automat programabil (AP). A fost dezvoltată o structură de date specială pentru a avea acces uşor şi a gestiona paletele prelucrate în patru staţii de lucru ale celulei. Au fost proiectate protocoalele complexe de comunicare în scopul de a permite interacţiunea AP-robot, protocoale care utilizează linii digitale I/E şi TCP/IP.

This paper describes a scientific approach for controlling a transport system integrated in a manufacturing cell using a single programmable logical controller (PLC). A special data structure has been developed to gain easy access and manage the pallets processed in four working stations of cell. There have been designed complex communication protocols in order to allow PLC – robot interaction, protocols that use both digital I/O lines and TCP/IP.

Keywords: PLC, intelligent manufacturing system, production activity control

1. Introduction

Throughout the world, the majority of automation systems use Programmable Logic Controllers (PLC) for manufacturing, transport and distribution industry. Since this concept has been introduced in 1960 until the present day, PLCs have evolved in an incredible pace, mostly by the addition of new functionalities, complex communication interfaces, high performance processors and by specializing on different areas of interest.

Control architectures that include PLCs are various, most of the time being determined by the application requirements specified by the end user. The simplest architecture is presented in figure 1 and it is used in simple applications that require controlling the electro-mechanical elements such as motors, actuators and valves, as well as a user interface.

The base architecture is it worth using only in processes that have small number of input output devices, because it depends directly on the PLC performances [1]. Also according to [2] one centralized architecture is not applicable to all circumstances and monolithic code reduces the scalability and reuse in case of a similar system.

---

[1] Phd student, Faculty of Automatic Control and Computer Science, University POLITEHNICA of Bucharest, Romania, e-mail: andrei@cimr.pub.ro

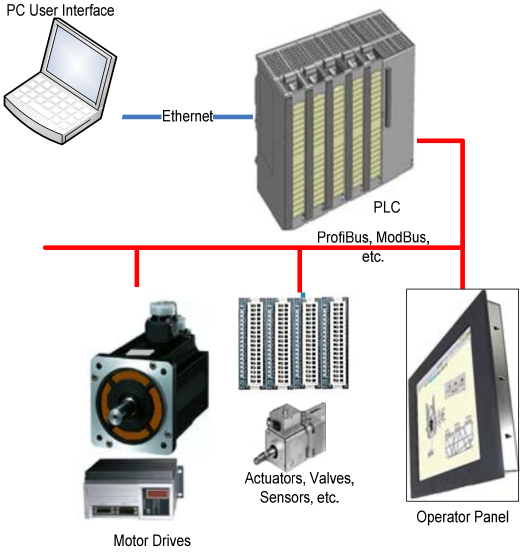

Fig. 1. Base Architecture of a PLC system

Complex applications that have thousands of input/output variables require more complex system architecture in which are involved more PLCs, each having under supervision and control, a process area. The PLCs are linked in a network in order to be able to communicate the parameters of the elements in the area they control[3].

From the point of view of the whole system control architecture traditional networked assembly structures are divided into two categories. The first one is called hybrid architecture and it allows the data exchange and co-operation between lower-level controllers. Also, in this case, a top level controller, usually called master, initiates all the activities and then the subordinates cooperate to execute them.

The second type of architecture is called heterarchical and it is formed by a group of independent entities named agents. The agents bid for orders (execution of operations) based on their status and future workload [4]. There is no more master-slave relationship and because of decentralization the agents are allowed complete local autonomy and the system is able to react promptly in case of a resource failure, new customer order, etc. [5], [6].

The new problem raised is that the optimization of a batch is improbable due to the different specifications and features of products (orders) that lead to system unpredictability [7]. In order to face break-downs as a base requirement [8] the networked assembly structures use multiple-LAN communications for saving production data and automatic re-planning [9].

2. The Control Problematic

The manufacturing cell consists of five robots and one conveyor transport system. The robots are able to communicate with the PLC through digital Input/Output lines and also by the means of Ethernet TCP/IP, being able to exchange information about their status and information regarding the manufacturing process of the pallet that is in the post. The stoppers existent in the system can block the movement of the pallets, some of them being related to sensor heads belonging to the identification sub-system. The lifts have the task of transferring the pallets on the main conveyor loop to the robot conveyor belts that take the pallet to the working stations. The conveyor system is controlled by a PLC that enables all mechanical actions to be taken in the transport system and as seen in figure 2, is the core component of the entire cell. It has to take the data necessary for production from a planner, and then needs to control al the mechanical elements of the transport system. It also facilitates the communication between all holons involved and has to monitor and track pallets throughout the system by the means of an identification sub-system and finally it has to communicate and give commands to the executing robots[10].

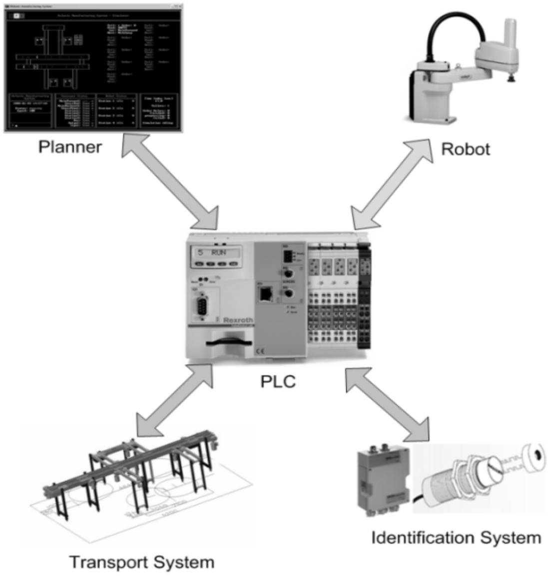

Fig. 2. Transport System Architecture

When the pallets enter the system, at the entry point a unique RFID code is written on the memory which is later used for both management and tracking. Each pallet follows a previous planned production and has to stop at one or more working stations in order to have the manufacturing operations done. Once all operations are complete, the pallet then is transported to the exit of the system. A batch can contain only 256 pallets due to the hardware limitations of the identification system that has eight bits available for the RFID code. In order to maximize the load of all available machines, an offline planner creates an optimal schedule.

The main criteria for planning the operations, is that the robots involved in the manufacturing process should have a minimum idle time. For this to be possible, the conveyor system should never be blocked by any product that is waiting to be processed by a robot. If the transport system is not overloaded, the robot station can always be reached without waiting periods of time.

In order to have this way of functioning there have been two constraints applied to the system:

- only four pallets can be in the system at any given moment of time;
- the products exiting from a working post have priority over the ones moving on the main loop of the transport system.

3. Solution design and implementing the order holon

The PLC software architecture it is described in figure 3. The data from the offline planner enters the system through the OPC communication module and then it is processed by four decisional units corresponding to each of the four working posts.

The decisional units control has three main functions: pallet identification – the unique code it is read and the system points out the next operation, robot job negotiation – the plc negotiates one last time with the robot the next operation that the product needs, and mechanical action management – takes al mechanical actions necessary to complete the movement of the pallet.

The failure management unit interrogates the robots and updates their status to both the user and the planner while the transfer unit takes the pallet from one conveyor belt to the other.

The application has to convert the data coming from the planner into mechanical movements of its constituent elements and, the first step in solving the control problem is to choose the way in which this data is stored. Having taken into account the way a product is developed it was decided to use this structure:

```
TYPE datemasina :
STRUCT
post:BYTE; (*number of the robot working post*)
operatie:BYTE;(* a code representing the operation done at this post*)
timpmin:WORD;(*minimum amount of time necessary for completion *)
timpmax:WORD;(*maximum amount of time for completing  operation*)
raport:BYTE;(*a small report about the result of the operation*)
END_STRUCT
```

END_TYPE

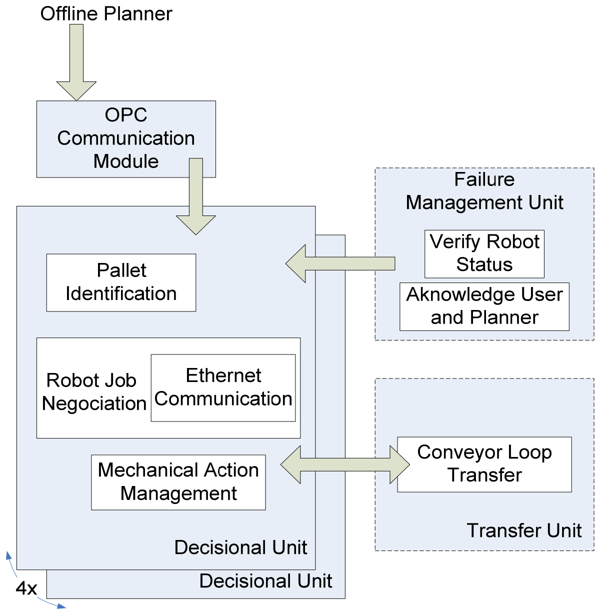

Fig. 3. Software architecture design

This structure is repeated for each operation that the product has to pass . It was decided that 16 structures are sufficient to completely describe a product and due to limitations of the system there can be no more then 256 products in one batch. In order to have access to the data in the array one needs two indexes: the first index is the product number (0-255) and the second is the number of operation (1-16).

The data structure necessary for the information regarding the production is very large 256 (products) x 16 (operations) x 5 (elements in a structure) = 20480 items and the limit imposed by the OPC standard is 15000 [11]. For this reason it was decided to send one product at a time over the OPC, meaning 16 structures of 5 elements and after 256 cycles the whole production is transmitted.

3.1. Theoretical backgrounds

When dealing with complex systems having a big number of inputs and output signals having to be controlled by a single PLC, a modern solution for software design must take into consideration the following issues:

- PLC must be chosen such as to support advanced software development kit, containing most of the standard programming languages for PLC's, like Sequential Function Chart (SFC), Ladder Diagram (LD) or Structure Text (ST)[12];

- The project should contain both graphical sequential programs and cyclic programs (for actions needed to be performed at every PLC cycle)[13].

In most of the cases the project should contain several programs that will run simultaneously, every program managing usually one action device (like relays, actuators and so on) or managing other activities like communication or timing management.

3.2. Project Structure

Following the guidelines previously described, a Bosch PLC together with Indralogic development software was used to accomplish the task. The project consists of 34 programs and 3 functions. All these can be divided into 9 categories: Stopper, Lift, Transfer, Communication with Robot (I/O), TCP data send to Robot, Write Code on Product, Time Keeper, User Interface, OPC Communication.

The “Stopper” type program has as objective the control of the mechanical device with the same name. Its main function is to stop products on the main conveyor belt and also include the decisional units. According to the planner data they can deviate the necessary pallets towards the working posts.

The “Lift” type program has as objective to take the pallet from one conveyor level to next. Once it has reached the work place it signals the robot that the operations can commence. Also it must lower the product from the robot work-post to the main loop.

The “Transfer” type program must control a double lift and the preceding stopper, its function being to transfer the product from one main conveyor belt to the other.

The “Communication with Robot” type program implements a status interrogation protocol over the digital input/output lines for all the resource controllers and has to decide whether the controller is working.

The “TCP data send to Robot” type program sends over the Ethernet the robot request operation code and waits for the answer using the three functions for the TCP communication directory.

The “Write Code on Product” type program has to stamp the unique identification code on the product memory capsule.

The “Time Keeper” type program measures the production time in 500ms units. As a second function, the program must reinitialize the system when a batch is finished.

The “User Interface” type program has two main functions. The first function is to control the graphic elements and show data in an easy to understand format. The second function is to permit the manual control of some of the conveyor system elements.

The “OPC Communication” type program must control the data flow from the planner to the transport system via the opc interface.

4. PLC Communication Interfaces and protocols

The communication between PLC and Robots is based on a dialog type model. According to this model, any communication protocol is initiated by the PLC, the Robot executes the orders and answers only in the mode depicted by the protocols.

The communication protocols that will be implemented must determine the robot status (online or offline), to coordinate the robot operations over the product and to transmit the codes of the operations and complementary execution report.

There are two communication interfaces at our disposal for implementing such protocol: digital I/O and Ethernet. The digital I/O lines are direct 24 volts direct current electrical line between the robots and the PLC and they are perfect for signaling and synchronizing operations. On the other hand they are much more costly and are not suited to transmit more complex data.

The PLC and the robots are interconnected via a switch into an Ethernet local network and are using the TCP/IP protocol for correct routing of direct messages and data. Although the network based on UPnP presented in [14] would make an excellent choice, due to the necessary complexity of the implementation it was decided to use a simpler special designed protocol.

The communication protocol will be consisting of two distinct parts:

- Status Interrogation – this protocol detects the online / offline status of the robots
- Synchronization – this protocol implements the robot task execution

4.1. Status Interrogation Protocol

One of the most important actions the PLC must take, is to determine which resource is online and when a robot changes it’s online / offline status. This is the main task of the Status Interrogation Protocol and due to its high redundancy requirements it was implemented using only digital I/O lines:

- From the PLC to the Robot Controller – “Request Status” through this line requests the Controller to signal that he is online;
- From the Robot Controller to the PLC – “Acknowledge Status” through this line the PLC waits for the answer from the Controller

The robot must always replicate the signals sent from the PLC in given period of time (usually 500miliseconds). Not answering rapidly enough determines the PLC to consider that the robot is offline or faulty if previously the robot had answered.

4.2. Synchronisation Protocol

This protocol must implement both PLC – Robot synchronization and job codes transmission. So, it is necessary to use both I/O lines and TCP/IP interfaces. From the PLC to the Robot Controller we require two signals: "Request_Job" - signals the Controller that the PLC needs a job to be executed; "Pallet_In_Position" - signals the Controller that the PLC has brought the pallet in the working position and that job execution can be commenced.

From the Robot Controller to the PLC there are necessary also two lines: "Ready" - signals the PLC that a job is in execution; "Job_Done" - signals the PLC that the current job has been executed.

At last the bidirectional (PLC ↔ Controller) link is done via the Ethernet interface using TCP/IP. The PLC transmits the job code, the Controller sends a job acceptance reply and (if the job was accepted), transmits job execution report upon job completion.

The protocol runs as follows:

- the PLC detects that Ready is at low level so it sets Request_Job to high level and transmits the job code over TCP/IP and the Controller reads the TCP/IP code and evaluates if it can execute the job: if he can execute the job then it will send the job acceptance code, if not it will send a job reject code; if the job is rejected then communication stops (we assume that the job is accepted)
- the PLC brings the pallet in the working position and sets Pallet_In_Position to high level and the Controller sets Ready to high level (the PLC resets Request_Job to low level) and begins executing the job
- upon job execution, the Controller sets Job_Done to high level, Ready to low level, and sends the job completion report over TCP/IP
- the PLC takes the pallet, resets Pallet_In_Position to low level, the Controller resets Job_Done to low level and is ready to recommence the protocol

The protocol's evolution over time is also presented in figure 4 and the intervals stand for:

- T1 – interrogation time of part supply and/or workstation equipments in order to reach to a job acceptance / rejection conclusion
- T2 – pallet transportation time from the main conveyor to the workstation
- T3 – job execution time

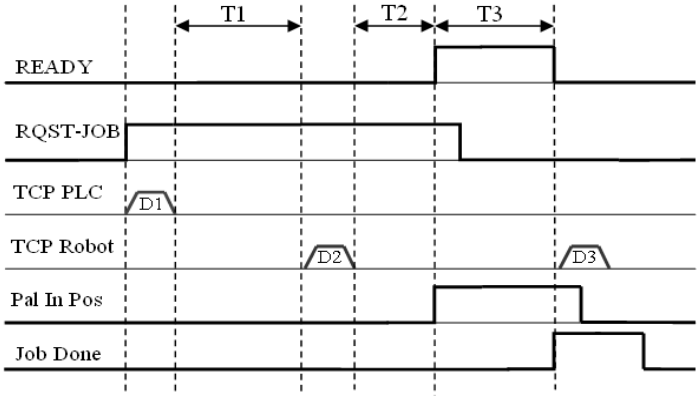

Fig. 4. Communication signals

5. Conclusions

After the theoretical design of the transport control system an evaluation of the solution was made and it was concluded that it fulfills all requirements needed to drive a manufacturing cell. Implementing the control solution involved also additional equipment such as vision guided robots, automatic feeding of bulk parts, CNC control for milling operations and doing so it was observed the capacity of the multitasking structure to execute the necessary tasks without interrupting the PLC communications protocols.

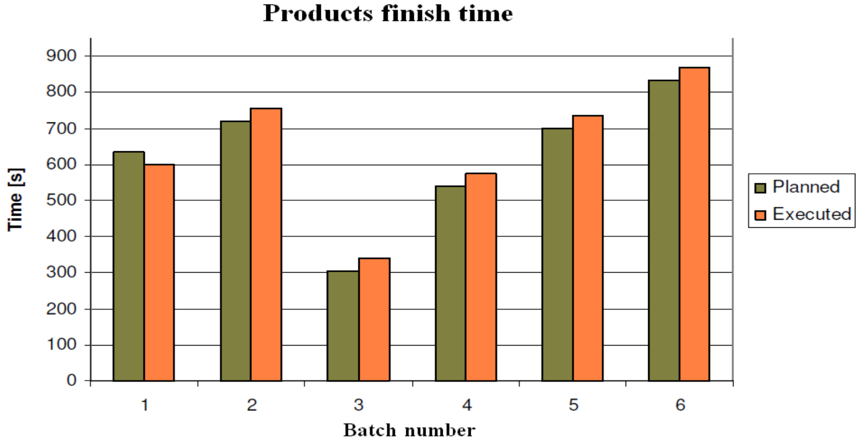

Fig. 5. Product finish times

After the solution programming was complete there was a supplementary task added that the PLC reports the start time and finish production time for both the planned and executed values. The results can be seen in figure 5 where the theoretical finish production times from the planner differ from those obtained on the system. The main reasons for that are: the mathematical model of the transport system taken into account by the planner is slightly different from that of the real system and there are delays due to the mechanical elements involved (mainly frictions on the conveyor loops). Overall production time is very close to the desired behavior of the system and the contributions made by the author: the development of the special data structure and the complex communication protocol have fulfilled their purpose.

REFERENCES

[1] *L. Payá, J.M. Azorín, N.M. García, J.M. Sabater, C. Pérez, R.P. Ñeco*, Process Control and Supervision through Internet, International Conference on Engineering Education July 21–25, 2003, Valencia, Spain

[2] *G. Black, V. Vyatkin*, Intelligent Component – based Automation of Baggage Handling Systems with IEC 61499, Theoretical Aspects of Software Engineering Conference (*TASE 2008*)

[3] *V. Realinho, F. Fidalgo, P. Silva*, "Low Cost Distributed Architecture for Supervising Systems", Current Developments in Technology-Assisted Education, 2006

[4] *G. Morel, H. Panetto, M. Zaremba, F. Mayer,* Manufacturing enterprise control and management system engineering: Rationales and open issues, IFAC Annual Reviews in Control, 2003

[5] *S. Rahimifard*, Semi-heterarchical production planning structures in the support of teambased manufacturing, International Journal of Production Research, 42, 17, September 1 2004, 3369-3382(14), Taylor and Francis Ltd

[6] *J. Barata, L.M. Camarinha-Matos, M. Onori,* A Multiagent Based Control Approach for Evolvable Assembly Systems, INDIN'05 - 3rd IEEE International Conference on Industrial Informatics 2005, pp. 478-483

[7] *J-L. Chim, D.C. McFarlane,* A holonic component-based approach to reconfigurable manufacturing control architecture, 11th International Workshop on Database and Expert System Applications, 2000, pp.219 - 223

[8] *T. Borangiu, N. Ivanescu, S. Raileanu, A. Rosu*, Vision-Guided Part Feeding in a Holonic Manufacturing System with Networked Robots, RAAD 2008, Ancona, Italy

[9] *R.F. Babiceanu et al*, Framework for control of automated material-handling systems using holonic manufacturing approach, Int. J. Prod. Res., 42, 17, Taylor & Francis, 3551-3564, 2007

[10] *M. Fletcher,* Building holonic control systems with function blocks, Proceedings, 5th International Symposium on Autonomous Descentralized Systems, pp. 247-250.

[11] The OPC Foundation", [On-line] available at: http://www.opcfoundation.org

[12] *V. Gourcuff, O. Smet, M. Faure,* "Efficient reprezentation for formal verification of PLC programs",8th Internationa Workshop on Descrete Event Systems,2006, pp.182-187

[13] *S. Gerksic, G. Dolanc, D. Vrancic, J. Kocijan, S. Strmcnik, S. Blazic, I. Skrjanc, Z. Marinsek, M. Bozicek, A. Stathaki, R. King, M. Hadjiski, K. Boshnakov,* Advanced control algorithms embedded in a programable logic controller, Control Engineer Practice,**Vol. 14**, Issue 8, August 2006, pp. 935-948

[14] *G. Veiga, J.N. Pires, K. Nilsson,* "On the Use of Service Oriented Software Platforms for Industrial Robotic Cells", IEEE International Conference on Robotics and Automation (ICRA2009), Kobe, Japan.