Visualization of a production line controlled by a PLC

(Car production)

Nagy Zsolt – F6ZKWV

Automatic Manufacturing System Project I.

Contents:

1. Introduction
2. Requirements
3. Features
4. Functional description
   1. Input
   2. Output
5. Architecture
   1. Block diagram
   2. HW Configuration
   3. Communication
   4. Interface
6. Implementation description
   1. Hardware
   2. Software
7. Verification against requirements
8. Time plan and milestones
9. Final demo
   1. Visualization
   2. Controlling
10. Conclusion

Introduction:

I have chosen this kind of topic because I am interested in PLCs and I am working in this field.

The name of my topic is Visualization of a production line by a PLC.

This kind of visualization presents the animation of a car production that is controlled by a Simatic S7-300 PLC or S7-1500.

Requirements:

The **first requirement** is that the pressing cylinder have to form a car from a raw material (cube) which is coming from a pre-processing unit.

The **second requirement** is that a spray booth has to paint the car.

The **third requirement** is that the painted car has to be located in a warehouse which will be chosed by according to the colour of the car.

Features:

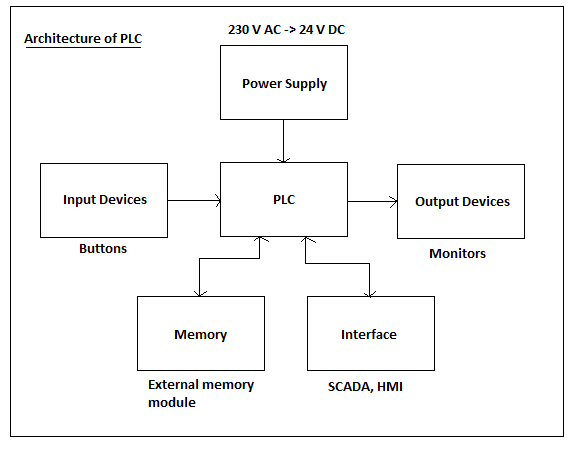
First step is a material coming from the pre-processing unit to the place where a pressing cylinder is going to form a car from the material. Then when this has happened then the car moves to the spray booth where the car will be painted (3 colors:red, green, blue) randomly. After that the car goes to the warehouse but the storage depends on the color of the cars because it will be 3 different colored warehouse.

Functional description:

* **Input**: The starting button on the production line will be an enable input.
* **Output**: We can see the whole procces on the screen of the PC.

Figure of the setup, architecture:

* **Block diagram**:



* **Hardware configuration:**

The hardware configuration is a very important part of the project because if you use a PLC without the hardware configuration you can not run the process of your project. Therefore you have to choose the appropiate devices and networks connecting to each other physically and on the PC too. However if you do not have a PLC that is not a problem because you can also run your program in PLC simulator.



* **Communication**: **Profinet/Industrial Ethernet**

It also helps businesses operate more successfully. Providing maximum flexibility, it offers you more freedom for tailored machine and plant concepts. Its high-level efficiency lets you optimally utilize available resources. The unique performance of this industrial Ethernet standard enables highest precision and product quality. Your advantage: a sustainable increase in productivity.

* **Interface: HMI (Human Machine Interface)**

SIMATIC HMI is optimized to meet your specific human machine interface needs using open and standardized interfaces in hardware and software, which allow efficient integration into your automation systems. From configuration software, to software for machine-level visualization, all the way to powerful SCADA systems with plant intelligence, to make HMI as efficient as possible – with more flexibility, transparency and openness.

**Used PLC: Simatic S7-1500 : CPU 1515-2 PN**

****

**Working process:**

The CPU contains the operating system and executes the user program. The user program

is located on the SIMATIC memory card and is processed in the work memory of the CPU.

The PROFINET interfaces on the CPU allow simultaneous communication with PROFINET devices, PROFINET controllers, HMI devices, programming devices, other controllers and other systems. CPU 1515-2 PN supports operation as an IO controller and I-device.

As an IO controller, CPU 1515-2 PN sends and receives data and signals from the connected IO devices within a PROFINET IO system. You can operate the CPU with a

maximum of 256 IO devices, of which a maximum of 64 may be IRT (Isochronous Realtime) devices.

In the "I-device" (intelligent IO device) function, CPU 1515-2 PN not only controls its own central modules, but also acts as an I-device, exchanging data with a higher-level IO controller. CPU 1515-2 PN thus fulfills the role of an intelligent pre-processing unit for sub-processes.

Implementation description:

* **Hardware (in the visualization)**: There are sensors (14-16), warehouses (3), spray booth (1), conveyor belt, color tank(1), motors(8), material handling equipment (slider)(5-6), machinig tool (pressing cylinder) (1) pre-processing unit(1).
* **Software**: TIA Portal V13 SP1

Step7 Professional V13 SP1

WinCC Professional V13 SP1

WinCC SCADA Runtime V13 SP1

My project has two parts. First the visualization part and then the controlling part. The visualization part will be do in the WinCC and the controlling part in Step7.

Using SCADA HMI.

**TIA Portal:**

For Siemens PLCs, Siemens has a development environment which name is Totally Integrated Automation in abbreviated form TIA Portal. It seamlessly integrates controllers, distributed I/O, HMI, devices, motion control and motor management into a single engineering environment. Thanks to its common data management and the smart library concept, comprehensive software and hardware functions efficiently solve all automation tasks.

**WinCC** is the software for all HMI applications ranging from the simplest operation solutions with basic panels to SCADA applications on PC-based multi-user system.

**STEP7** is the ultimate engineering tool for configuration and programming for all SIMATIC controllers.

**SCADA RT**: The runtime functionality is determined by the features of the HMI device used, such as available memory capacity or the number of function keys and can be extended by means of options.

Some options are only available for panels above a specific class, while others are integrated.

Industry-standard HMI functions are part of the basic equipmentof the system::

Fully graphical visualization of the process sequences and statuses, including dynamizable graphics, bar graphs, and gauges.

Operating the machine or plant via an individually configurable operator interface, e.g. with buttons, switches, and sliders.

Reporting and acknowledging of events.

Logging of measured values and alarms.

Reporting of current process data and recorded log data.

User administration including their access rights.

Verification against requirements:

Each small parts is tested step by step to avoid errors.

Error handling (including negative errors).

Validation test to prove that the process is good enough.

PLC programming is tested with PLC simulator and also with the PLC.

The visualization can be tested with the correct PLC program.

Gantt chart:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 8th of february | 15th of february | 22th of february | 29th of february | 7th of march | 14th of march | 21th of march | 28th of march | 4th of april | 11th of april | 18th of april | 25th of april | 9th of may | 2nd of may |
| Weeks: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Choosing the idea | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Specification |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Extended specification |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Starting the visualization |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Visualization |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Visualization |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Finishing visualization |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Starting the controlling |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Controling |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| First demo |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Upgrading and checking |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Final demo |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Milestones:

1.HW Config 4. Presentation

2.Visualization 5. Final demo

3.Controlling

Final Demo:

* **Visualization: Components:**

**Pressing cylinder with**

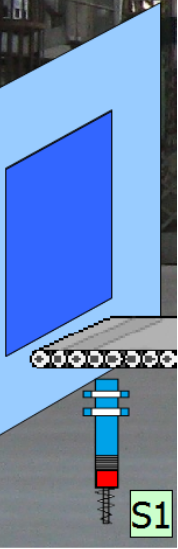
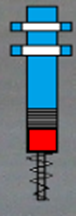
**Color Tank top/low position sensors**

** **

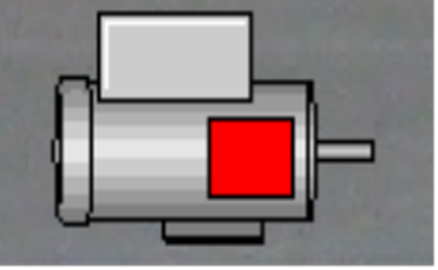
**Working process screen**



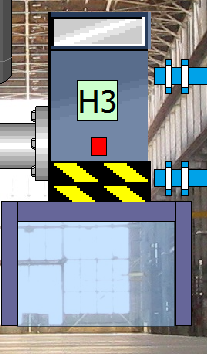
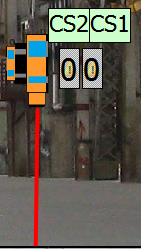
**Pre-processing unit sensor**

** **

**Motor**

****

**spary booth warning light**

** **

* **Controlling part:**

The code had been written here in STEP7.

I have used two types of programming languages.

LAD and SCL.

I have divided my program codes into several parts (networks). One of them is written in LAD(controlling) and the others in SCL(graphical part).

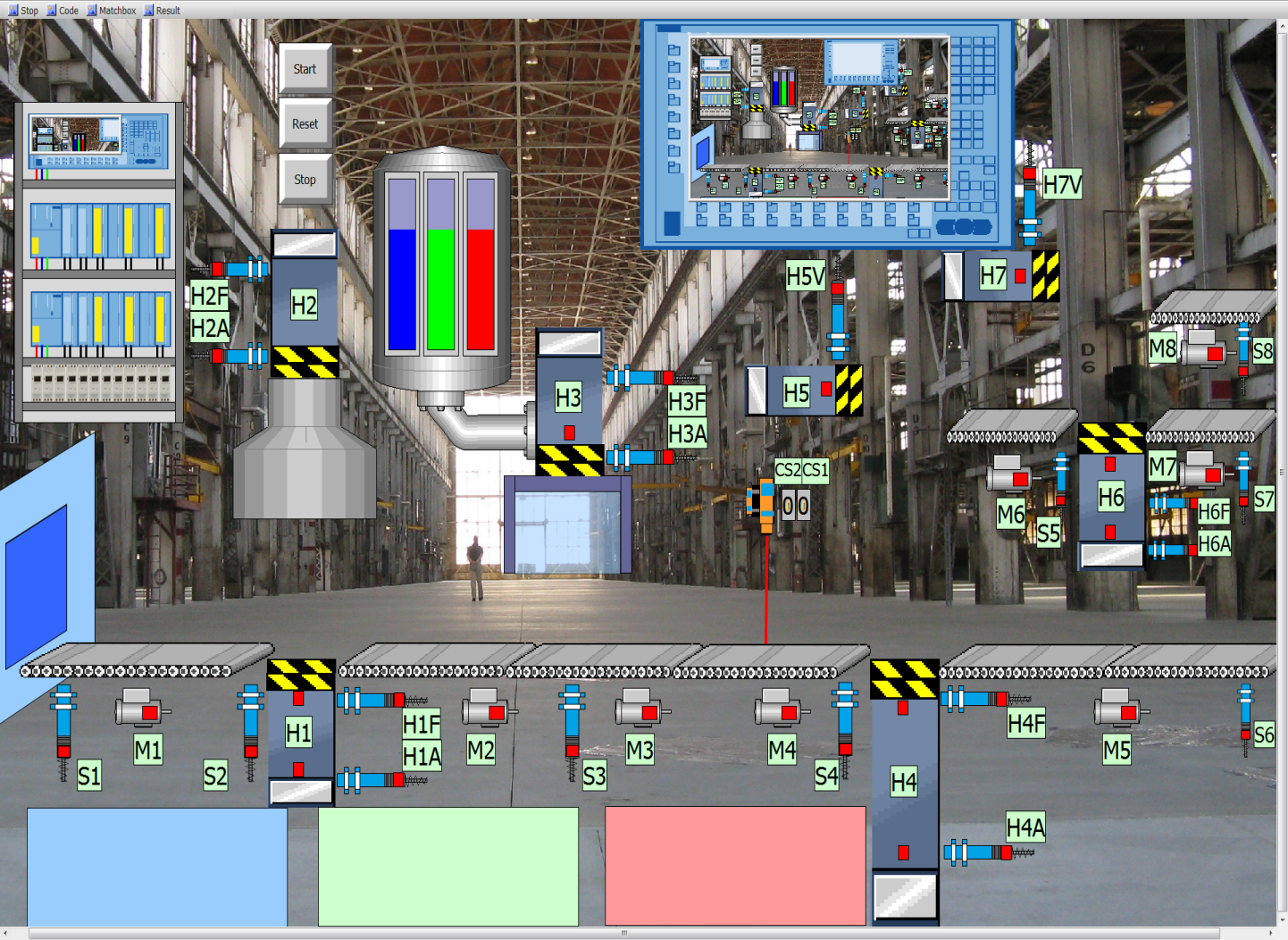
I have done the controlling part by taking to pieces.:

The first one was from pre-processing unit until pressing cylinder, the second one from pressing cylinder until spray booth, and the third one from spray booth until warehouses.

Problems: in early phase with the third part but later solved.

By the end of my project I had a lot of program blocks because as I mentioned before that I wrote the project in more blocks.

**Final Version**



**The whole process:** A material coming from the pre-processig unit then the sensor one is activated, when the sensor is activating the motor one is switch on and drives the conveyor belt, so the material can go forward to the pressing cylinder. When the material is between motor1 and sensor2, then the sensor2 has to be activated and when its activated, then the H1A and H2F top and low position sensors have to be activated so the pressing cylinder(H2) and the pusher(H1) will take into motion and make a car from the material. When it has happened the H1F and H2A top/low position sensors return to starting position and switch on the motor2 and drives the conveyor belt at the same time. When motor2 is on, then sensor3 is activating and H3F top position sensor will enable the color tank to paint the car but the color will be painted by the H3 pressing cylinder. When this process is finished the H3A low position sensor will activate and runs the motor3 which will drive the conveyor belt again. After that we arrived to the warning light which will decide that our car has painted or not if it has painted corrrectly then the car goes further if not then it is an error and the process will start again from the begining. Furthermore, when the car meet the expects then motor4 will switch on and drive the conveyor belt. When this has happened sensor4 will be activated and checks that the car is red or not if it is red then H4A will activate and H4 will move between sensor4 and H4F to get the chance for the car to go further to the next conveyor belt which is drived by motor5. After that sensor6 will be activated and open the red warehouse to store the made car. But if the car is not red, let say green then H4 pusher will push the car by one level until the next conveyor belt which is drived by motor6 but the H5 pusher will push the car to the conveyor belt. When this has happened sensor6 will be activated and checks that the car is green or not if it is green then H6A will activate and H6 will move between sensor6 and H6F to get the chance for the car to go further to the next conveyor belt which is drived by motor7. After that sensor7 will be activated and open the green warehouse to store the made car. But if the car is blue then H6 pusher will push the car by one level until the next conveyor belt which is drived by motor8 but the H7 pusher will push the car to the conveyor belt.

When it has happened then motor8 will activate the conveyor belt and sensor 8 will turn on and open the blue warehouse to store the made car.

Conclusion: My opinion is that the begining of the project I was a bit confused and I cannot imagine that how can I solve the visualization but later when I get the hang of it and carefully thought about then it did not seem to be so confused at the end.