

SYSTEM REMOTE CONTROL FOR A RUBBER DAM

Filippo Vecchi

Genova University
Genova, Italy
filippovecchi@hotmail.it

Abstract

To automate a process and to make it remotely controllable is more and more required in industries. In recent years, personal computers assumed an important role in technology, infact PCs provide an ideal hardware platform for controlling and displaying the process data when they are connected to an operating system "Real time". Using modern and flexible graphical programs (SoftPLC) it is possible to create an automation system that largely surpasses the performance of the most powerful PLC, utilizing flexibility and connectivity of PCs.

The application where has been applied this technology is the inflatable rubber dam.

It was designed and assembled a scale model with automation system and HMI accessible from a web page.

I. INTRODUCTION

A. What is a Rubber Dam

Rubber dam is a remotely operated gate for water control. It is comprised of a nylon reinforced rubber body secured to a concrete foundation with a unique anchor mechanism. When the body is filled with air from a blower, it rises to block water. When the air is released through an exhaust valve, the body lays flat on its foundation to permit the free flow of water.

Rubber dam can be installed in span over 200 meter long and can be produced in heights of up 6 meters, the rubber body has a life of 30 years during which time it's free from routine maintenance.



Fig. 1. Rubber dam axonometric section. This figure shows how rubber body is secured to a concrete foundation by its anchor line.

B. Design and implementation steps

Rubber Dam model design and implementation can be divided in four steps. First of all, it was analyzed an existing installation to identify project specifications (*phase one*), than it was defined hardware and software architecture of our model (*phase two*).

Third step was consisted in the control logic and automation system manufacturing and its installation in the rubber dam model (*phase three*). Finally some tests were done on the model (*phase four*).

II. EXISTING INSTALLATION ANALYSIS

Analysis was done in a ENEL hydroelectric power plant located in Narzole (CN) Italy; plant's name is "Isorella sul Tanaro".

In this area two Kaplan turbines (*35 MW each one*) are installed, rubber dam blocks water from Tanaro river to pipe water to the turbines.

Tanaro installation has the following characteristics:

- manual inflating and deflating. This configuration needs the presence of man;
- none water level measurement;
- emergency mechanical deflating activated in case of river flood.

III. PLANT'S AUTOMATION STUDY

A. Goal of automation

Installation's variables are: pressure inside the dam and water level of the river. Through an automation system we would control pressure inside dam between two values (*minimum and maximum acceptable pressure*), identify two water levels and view installation state by a remotely accessible web page.

B. Architecture basis

Before seeing complete system architecture, it's opportune to consider a simplified architecture.

The core of the project is constituted by the SoftPLC, software that transforms an industrial PC in a process controller with all the functionality (process

management, I/O control) of a traditional PLC.

By using a SoftPLC there are the following advantages:

- 1) realization of a very complex application;
- 2) possibility to take advantage of PC potentiality, flexibility and connectivity;
- 3) control of a greater number of distributed I/O;
- 4) control of a greater amount of data;
- 5) hardware saving.

SoftPLC has to be able to interface itself with both the field (fieldbus and distributed I/O modules) and HMI, Internet. In particular the SoftPLC used in this application communicates through standard communication OPC [1], a standard based on client-server model, that offers some important advantages as safety, reliability, speed of data access and low cost of interface.

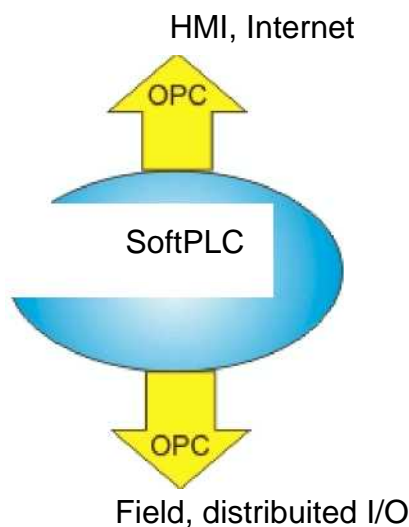


Fig. 2. SoftPLC interaction

In order to obtain an agreement between control planners and PLC and SoftPLC producers, it has been introduced the IEC 61131-3 norm [2] that defines standard programming of control automation devices. This norm defines five programming languages that allow to program logic control in SoftPLC.

The five languages are:

- 1) SFC: Sequential Function Chart;
- 2) LD: Ladder Diagram;
- 3) FBD: Function Block Diagram;
- 4) IL: Instruction List;
- 5) ST: Structured Text.

IV. HARDWARE AND SOFTWARE ARCHITECTURE

Figure 3 shows Hardware and Software design.

The core of the project is constituted by Infoteam OpenPCS [3], the SoftPLC in accordance with IEC 61131-3 [2], where the logic control of the system is placed. Logic control cyclically elaborates a series of data coming from field and shows

installation state to user.

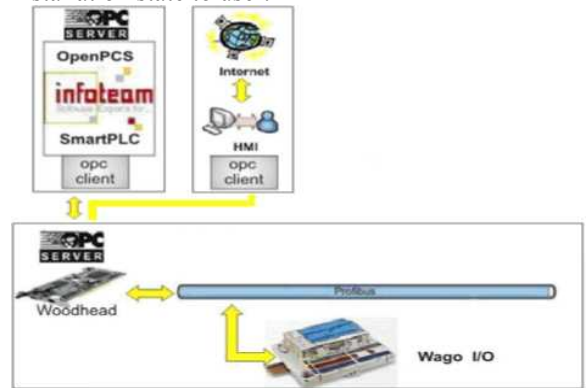


Fig. 3. Hardware and Software design

OpenPCS needs software, Infoteam SmartPLC/OPC [3] in order to communicate with the field and with the user using standard OPC [2]. SmartPLC/OPC, beyond executing code of logic control, is as OPC server and as OPC client.

Fieldbus is the mean that allows to send and to receive data from I/O distributed modules. In our application we have used Profibus DP. For being able to use such fieldbus it's necessary to equip with appropriate board to install in computer. This board can be manage with OPC server that allow to "collect" all the information coming from the field. This OPC server communicates directly with SoftPLC's OPC client, realizing therefore the interface between logic control and field. The interface with the user is based on standard OPC too. Controller plant user can connect, also through Internet, to a web page (Human Machine Interface, HMI) in order to look at the installation state.

V. CONTROL LOGIC AND AUTOMATION SYSTEM MANUFACTURING

A. Introduction

Before describing how it is made the control logic, the automation system manufacturing and its installation in the rubber dam model, it's necessary to introduce a fundamental concept. In this application the process is on scale and is reproduced by a model but the control logic and automation system can be installed in a real process like "Isorella sul Tanaro" hydroelectric power plant.

B. Variables definition

The system that we want to automate, could be studied through the use of boolean variables because it can be represented with inputs and outputs characterized by two levels. One corresponds to a state of "true" (1), other identify the opposite state "false" (0).

The inputs and the outputs in the model are:

Inputs:

1. minimum pressure inside the dam (I0.0) $\rightarrow P_{\min}$.
2. maximum pressure inside the dam (I0.1) $\rightarrow P_{\max}$.
3. minimum water level (I0.2) (*also nominal water level*) $\rightarrow L_{\min}$.
4. maximum water level (I0.3) (*when water starts*)

to overtop rubber dam) $\rightarrow L_{max}$

Outputs:

1. blower (Q0.0).
2. deflation valve (Q0.1).

Each I/O has a relay contact, so when that contact is open the variable is "false", when the contact is close the same variable begins "true".

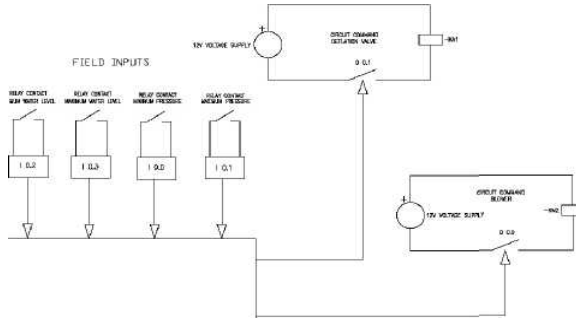


Fig. 4. Interaction between logic control and field

C. Logic control implementation

The logic control has been realized in OpenPCS [3] using "Function Block Diagrams", a programming language defined in IEC 61131-2 [2].

Two functions were implemented; first one to run blower, second one to open deflation valve.

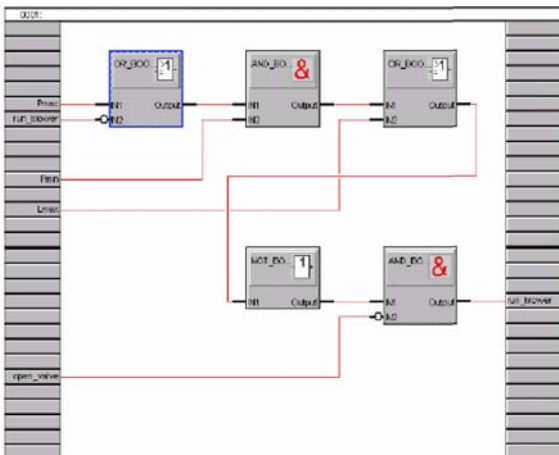


Fig. 5. FDB diagram.

This diagram shows logic control that command blower.

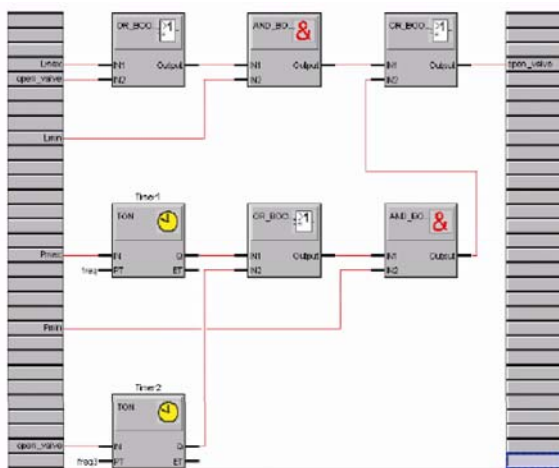


Fig. 6. FDB diagram.

This diagram shows logic control that command valve.

D. Logic operation (pressure control)

Blower gives pressure inside the rubber dam. Pressure must be stay between P_{min} and P_{max} , if pressure is between these values blower is stopped and deflation valve is closed.

When Pressure is less than P_{min} blower runs until pressure reaches P_{max} value.

If pressure is more than P_{max} for more than 30 seconds deflation valve opens and, with the same logic it closes when pressure is less than P_{max} .

E. Logic operation (level control)

When water level reaches L_{max} value, deflation valve opens and closes when water level is less than L_{min} .

To be more precise is necessary to underline that blower cannot run when deflation valve is open.

F. Fieldbus node

Now we analyze the components constituting fieldbus node. All the elements are parts of Wago I/O System 750 series.

- 750-602: Supply Module;
- 750-434: 8-channel digital input module;
- 750-434: 8-channel digital input module;
- 750-513: 2-channel relay output module;
- End module.



Fig. 7. I/O modules

G. HMI

Human machine interface has been realized with the software GraphWorX 32 (ICONICS product) which creates dazzling animated graphics.

GraphWorX 32 is an OPC Data Access client application, that means it can easily plug-n-play not only with ICONICS servers and components, but other 3rd-Party hardware interface drivers and software as well.

This interface is representative of P&ID. Instrumentation and actuators are green or red colored, it depends on the variables state. Green color means that variable associated to the element is "false"; so red color means that the same variable is "true".

In the following figure is possible observe a page from HMI.

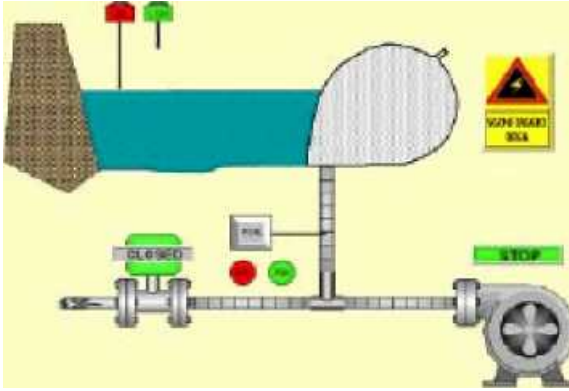


Fig. 8. HMI page

This page is in HMTL format, uploading this one on a web server, user can view it remotely through internet.

VI. RUBBER DAM MODEL CONSTRUCTION

Rubber dam model structure is wood made and covered, to waterproof it, by 1mm steel plate.

Actuators, instrumentation and the auxiliary field are located on a galvanized iron plate which is placed inside the structure. It was built also a small switchboard visible in the following image (fig. 9).



Fig. 9.

VII. FUTURE DEVELOPMENTS

A. Field side

In this installation all I/O are digital (we have only 0 state and 1 state), a possible system development could be made using analogical variables to look effective values of pressure and level water.

B. Web side

In this architecture user can also look at the system state.

To command manually outputs in the field, from remote web page, could be an interesting development.

REFERENCES

- [1] <http://www.opcfoundation.org>
- [2] Karl-Heinz John, Michael Tiegelkamp, *IEC 61131-3: Programming Industrial Automation Systems*, Springer.
- [3] <http://www.infoteam.de>