

GREENHOUSE MANAGEMENT THROUGH CAN NETWORK AND PLC ON A TOUCH SCREEN SYSTEM

Dr. R. Barbagallo

Computer Engineering Department ,University of Catania
Catania, Italy
barbagallosaro@virgilio.it

Abstract

This paper describes a distributed control system for an agriculture plant made up of three greenhouses, each one controlled through a single PLC. Each PLC must control the temperature and humidity inside a greenhouse through the suitable activation of a fan, irrigation system, heater and the opening/closing of some windows. PLCs are connected through a CAN bus to a central monitoring station with a graphic interface. The design and development of the interactive interface "Greenhouses Management" has been the more intense and difficult part of the job. Four principles have been adopted for a user-centered design: understanding the user, interactive design, constant measurement of the results achieved, iterative planning. The main objectives have been: to speak the language of the user, to minimize the quantity of memory required, to provide a continuous feedback and to help the users to recognize, diagnose and recover the errors.

I. INTRODUCTION

This papers analyzes and emulates all the operations which are required for the remote managements of three Greenhouses which are represented by three cards interconnected through a Can network. A graphical interface allows to control and monitor remotely the three greenhouses and set appropriate parameters. Every card has been connected to a PLC and the entire system communicates with a master node through a touch-screen PC TC1000. Each PLC, uses a firmware written in FBD language, and has the function to manage the systems which control the greenhouse, i.e. the heating, ventilation, opening and closing doors and the dehumidification system. The implementation of the GreenHouse System has required the use of PLCs, Microcontroller boards with communication capability, an industrial PC with touch-screen monitor, various sensors and actuators, and some spare electronic components. Starting from

these basic components, I have assembled a complex system coping with all the problems which come out from Hw/Sw integration and from the integration of all the devices. This is a complex activity, which allowed me to face with many problems:

- Study and design of a distributed control system to be implemented with the available components.
- Development of the PLC software and configuration of its interfaces.
- Development of the microcontroller software.
- Design and implementation of simple sensor boards.
- Configuration and management of the communication boards.
- Design and development of a suitable Human Machine Interface for the management and monitoring of the system.
- Testing of the system.

The paper is organized into three main sections: considerations on the System components, analysis of the entire system and finally the user's interaction with the system.

II. COMPONENTS

Microcontroller board

The microcontroller board is based on the use of an 8 bit PIC18F458 microcontroller from Microchip. Its pin layout is shown in Fig.1.

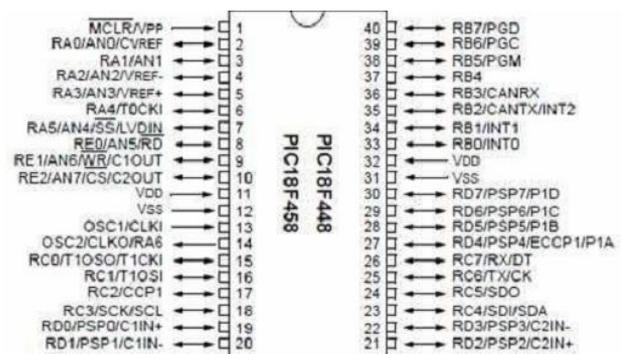


Fig. 1. PIC18F458 microcontroller pin layout

This microcontroller contains several resources:

- An 8 bit CPU.
- A flash memory where the program can be stored.
- A RAM Memory which can be used to store the variables used by the program.
- Several I/O lines. Some of them are digital and some are analog. Through this lines it is possible to provide a analog/digital data acquisition which can expand the capabilities of the PLC (which possess only digital I/O ports).
- Several auxiliary devices (clock generators, counters, timers, etc.) which can be used for driving external devices.
- A digital I/O line, compliant with the CAN data MAC Protocol.

All these features are synthesized in fig.2.

Device	Program Memory		Data Memory		IO	10-bit A/D (N)	CCP/ECCP (PWM)	I ² S/SPI		USART	Timers 8/16-bit
	FLASH (Bytes)	# Single Word Instructions	SRAM (Bytes)	EEPROM (Bytes)				SR	Master/PC		
PIC18F458	32K	16384	1536	256	33	8	2	Y	Y	Y	13

Fig. 2. PIC18F458 features

The PIC18F458 microcontroller has been chosen according to its versatility and simplicity of use. Software can be written directly into Assembly language or in C language. In the last case, a free programming environment MPLAB IDE provided by Microchip can be used. The microcontroller has been integrated into the board together with some support components such as:

- A MAX232 integrated circuit with the task of adapting the voltage levels provided by PIC18f458 with those foreseen for the RS232 standard. This allows the microcontroller to be connected to the serial port of a PC.
- An MCP2551 integrated circuit which implements the physical layer of the CAN protocol and allows several boards to communicate through the CAN bus.
- A 20Mhz quartz clock generator which provides a stable reference time for the synchronization of the various processes.

The board, with all the devices which are implemented there, provides a set of functionalities which are very useful for the programmer. It can be used for a wide range of different applications and allows a programmer to design and implement a distributed system, based on the CAN fieldbus, with a minimum effort in the hardware implementation. (fig 3) In the design of the device set, I devoted great attention to make the system flexible, in order to spend most of time for the solution of conceptual problems, software development integration of components rather than their physical implementation.

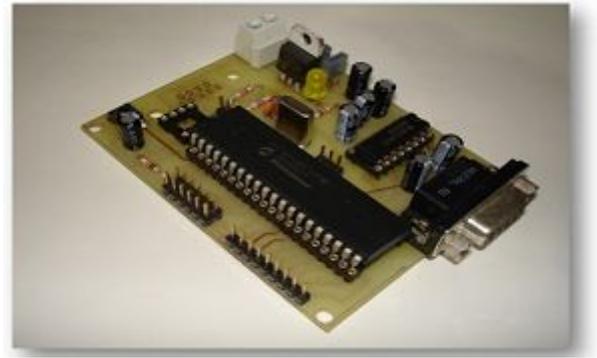


Fig. 3. Greenhouse Board: this is the heart of each greenhouse and is responsible for receiving data from sensors and marshalling data to PLC and PC

The presence, inside the microcontroller, of a CAN interface allows an easy interconnection of the various components in order to make a distributed system. CAN was initially designed for automotive applications, but in the last years it has been widely used for industrial applications, mainly in the automation area. A great push has been provided by CAN In Automation (C.I.A.) consortium whose task is to develop the use of CAN.

CAN is really great and is very suitable for:

- Simple to use: the CAN protocol is very simple and can be easily understood and managed by programmer. MAC protocol is implemented inside several microcontrollers by many manufacturers, so that it is possible to use the same communication protocol also with different hardware implementations.

Moreover, this protocol is very open and the programmers can set parameters and modify their configuration easily.

- Expandable: in theory it is possible to connect an infinite number of nodes. Practically speaking there are some physical constraints which limit the number of nodes, but in any case it is possible to implement networks with a high number of nodes without problems.

- Flexible: a multicast communication is used based on data identifiers. Up to 2032 different message identifiers are available, and each one can contain up to 8 Bytes. Nodes are not addressed explicitly this way allowing the user to add new nodes without requiring any Hw/Sw modification in the system.

Access to the bus is managed by a modified CSMA/CD protocol which allows the implementation of a priority based traffic scheduling also allowing the programmer to test and evaluate different communication policies.

- Reliable: some efficient mechanisms for detection and confining of errors are available. Programmers can monitor the system operations and make statistics on error rates in the network.

- Robust: CAN continues to operate even in the presence of faults in a node or a break in the line.

- Cheap: the cost of all components is very low and this allows to implement a new board at the cost of about 10 Euros.

- Fast: the maximum transfer rate is 1 – 2 Mbps which is enough high to support even real-time application.

However, the transfer rate is affected by the bus length. For a 40 meters bus length the max bit rate is 1 Mbps. In case of longer distances it is necessary to reduce the bit rate.

PLC

PLC is the basic element in an automation system. It implements the control algorithms through the data acquisition from input sensors and providing output commands to the actuators. The choice of a PLC suitable for this project has taken into account the following points:

- PLC must be simple to use without requiring long training, but enough sophisticated to allow the implementation of complex applications.

- It must use several programming languages.

- It must have an efficient emulator which allows the user to test the software before it is downloaded in the PLC memory.

- It must be easy to interface with other devices.

- It must have a low cost.

All these features have been identified in the SR3101BFU PLC from Schneider electric. This device is characterized by 6 digital inputs and 4 Digital outputs, it can use both FBD and LD programming languages and is directly powered by 100-240V.A.C.

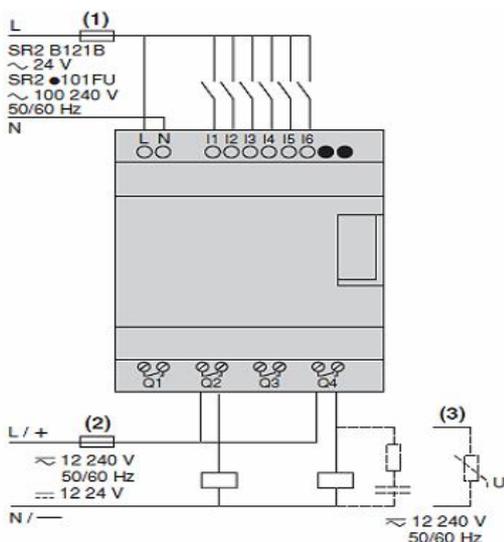


Fig. 3. PLC external connections schema

This PLC is suitable for small automation systems as it allows to implement simple control panels and it is widely used for small industrial plants. Programming of the device is possible through two well known PLC programming languages: Ladder (LD) and Function Block (FBD) languages.

Programming can be performed through some buttons on the PLC front panel or through a PC, by using the emulator software. In this last case, through the Graphic Programming Interface, programmers can quickly implement control algorithms based on the instructions and Function Blocks provided in the system. Moreover, they can make this work at home and simulate it in order to check the correct relations between inputs and their relevant Outputs and correct possible errors if any. Only when the test is OK programmers can transfer the software on the PLC and test it on the real system.

Touch-Screen Industrial PC

The last element of the system is a PC which can be used mainly for monitoring the control system. To this aim an industrial TC1000 PC from Schneider Electric has been chosen. This computer is equipped with a Touch-Screen monitor which can be used to implement graphic interfaces written in java or c++ languages. The PC is in such a way the device which integrates all components and transform them into a distributed control system. The research and design of new human system interactions ways which simplify the use of the system.



Fig. 4. PLC external connections schema

III. SYSTEM: MANAGEMENT OF A GREENHOUSE SET.

The project was divided in several tasks :

- Designing the distributed control system for an agriculture plant made up of three greenhouses, each one controlled through a single PLC.

- Each PLC must control the temperature and humidity inside a greenhouse through the suitable activation of a Fan, irrigation system, heater and opening/closing of some windows.

- PLCs are connected through a CAN bus to a central monitoring station with a graphical interface.

In order to perform the whole task:

1. Develop the control software inside each PLC.
2. Configure the CAN nodes.

3. Interface each PLC with can nodes in order to implement a distributed system.
 4. Design and implement some additional hardware to interface analog sensor to the digital inputs of the PLC.
 5. Design the graphical interface.
- The main part of the system are represented in the following block diagram:

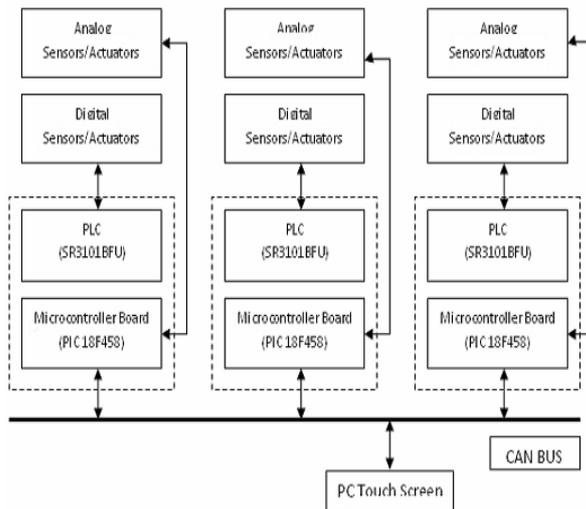


Fig. 5. Block Diagram of the Green House Project

In this design, one feature is worth to be outlined: the small PLCs possess only digital I/Os and are not equipped with communication ports. Their integration into a distributed system, where are present both analog and digital sensors/actuators, represented a further problem that the programmer must solve connecting them with the microcontroller boards. In fact, the microcontroller boards, as said, are able to communicate using the CAN protocol. In addition the microcontroller analog I/O port have been used to interface with analog devices. Then the analog signals are converted by the microcontroller ADC and sent to the PLC to be processed. The same happens in the opposite direction through the microcontroller DAC. In this way, the PLC and the microcontroller board, together, can be seen as a new object obtained by composition of two simpler objects, as in the object oriented programming paradigm. One of the nodes is used as Master, and was connected to the TC1000 industrial PC, where a control application allow a user to monitor and setup the system.

Implementation of the PLC Software

To implement the task performed by the PLC, I have used the FBD language and the main blocks which are shown in Table 1. By mean of these blocks it is possible to make simple control as those required for the Greenhouse. Input functions, shown in table 2, have been used in order to acquire from the sensors the information, required for the control. This point requires some additional information. As the PLC has only digital inputs, whereas the information provided by both temperature ad humidity sensors are analogic, have used the analog input of the

microcontrollers to read the values and then have changed them into digital values through the software implementation of a Schmitt trigger provided by a suitable FB of the PLC. At last, table 3, shows the output functions used to drive the actuators. Four actuators have been used:

	Boolean Blocks: through the implementation of suitable truth tables is possible to correlate input values with output values.
	Display Blocks: allow to visualize suitable messages on the PLC display.
	Compare Blocks: allow to compare input values with some treshold values.
	Number Blocks: allow to fix some treshold values.
	Timer Blocks: allow to modify the behaviour of the system during the day.

	Input 1: Power ON/OFF
	Input 2: Position sensor. Window is open
	Input 3: Position sensor. Window is closed
	Input 4: temperature Sensor
	Input 5: Humidity Sensor.

	Output 1: Window opening Motor ON/OFF
	Output 2: Fan ON/OFF
	Output 3: Irrigation ON/OFF
	Output 4: Heating ON/OFF

- A motor is used to open /(close) a window in the green house when the temperature or the humidity are too high/(low). Its position is monitored by the limit switch connected to Input 1 and Input 2.
- A fan is used to move the air when the temperature and humidity are too high. This prevent the formation of mould on the vegetables.

- An irrigation valve is activated when the ground humidity is too low.
- An electric heating system is activated when the temperature is too low.

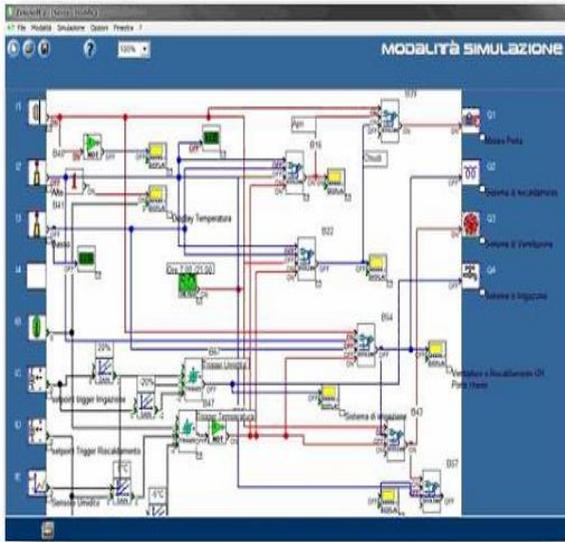


Fig. 6. FBD for the greenhouse control

Microcontroller Programming

The PIC18F458 microcontroller programming required several phases:

- Assigning the configuration word in order to set some parameters.
- Initializing some register and the CAN network. One node was configured as a Master, with the task to exchange, by means of the RS232 serial port, command/data with the monitoring station. The other nodes exchange data with the master node.
- Configuring I/O ports to acquire and digitize the values provided by the analog sensors.
- Configuring an I/O port in order to read the values which are present at both the Input and Output connectors of the PLC. This task is indispensable in order to transform a set of independent PLCs into a distributed system.

Monitoring Graphic Interface

The graphic interface has been developed on the Touch-Screen TC1000 PC, as a Java application. The interface required a suitable definition of the screen properties in order to exploit its touch screen capabilities. Some buttons have been implemented to set command for the system. The graphic interface is shown in fig. 7.

The graphic interface allows:

- Realtime monitoring of both the temperature and humidity of the greenhouse.
- Monitoring the state of the various actuators.
- To set, for each greenhouse, the threshold values for both temperature and humidity that, if exceeded, activate the control systems.

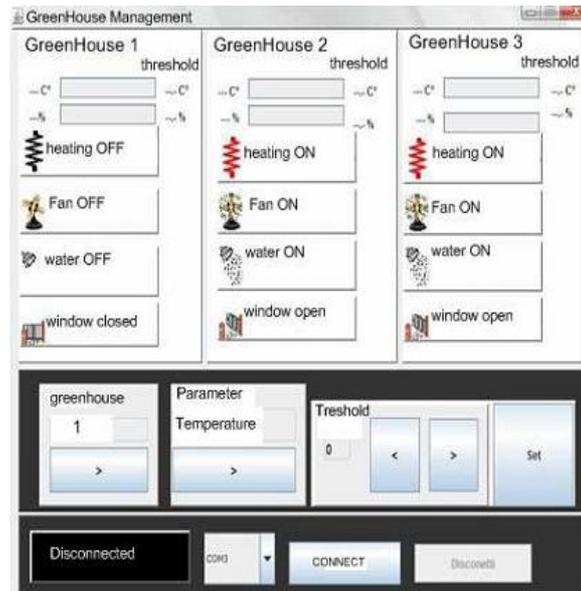


Fig. 7. FBD for the greenhouse control

IV. CONCLUSIONS: THE HUMAN MACHINE INTERFACE

I have developed a system for managing three greenhouses in order to provide people which operate in the agricultural area a safe and efficient system. A fundamental component of the system is the human-machine interface whose task is to allow an easy interaction with the system. The design and development of the interactive interface for "Greenhouses Management" has been the more intense and difficult part of job. During the development, over the 50% of the code has been devoted to the interface. The complexity of the design of the interface has depended on the variety of choices and design decisions to perform all possible tasks, often unpredictable. Four principles have been adopted for a user-centered design:

1. Understanding the user:

An explicit representation of the competences of the user and of the work to be done.

2. Interactive design.

A representative sample of users has been part of the design team during the development process.

3. Constant measurement of results achieved.

Since the first phases of the planning, human-computer system has been tested and evaluated.

4. Iterative planning.

The design has a cyclic nature: design, test, measure, redesign, until a human-computer system performance has fulfilled the purpose for which the system has been conceived. The development of the prototype of the three greenhouses has not only been characterized by the tools used but also from the refinement of design ideas, the improvement of the firmware of the PLC, of the software developed for the pic18f458s and the entire system. To design a system for managing greenhouses where hardware and software must be in accord with each other and

with the ideas of a specific user has been a complex task. The main objectives have been:

1. Speak the language of the user.

I used a terminology based on the language spoken by users in order to carry out their assignments. I used words, symbols, icons mnemonic and highly significant in order to facilitate the remembrance and recognition. The graphical interface developed for the PCTC1000 has been made easy and intuitive so that the user does not lose time or stop its activity. Animations and dynamic indicators of temperature and humidity have made the interface more lively and clear increasing the interaction and providing an instantaneous perception of the state of the system. In absence of the graphic interface the screen of the PLC has been programmed to provide the information necessary to monitor the ventilation system, heating, irrigation and opening or closing the doors.

2. To minimize the load of memory of the consumer. The recognition is preferred rather than the memory.

3. Provide constant feedback.

During the interaction with the system, it has been guaranteed immediate and effective feedback on the effects of the just finished action. In absence of the graphic interface the screen of the PLC has been used so that the user will be always informed on the effects of his actions on the system.

4. Help users to recognize, diagnose and recover the errors. Through clear and detailed error messages the user is allowed to know what the problem is and how to prevent or recover it.

REFERENCES

- [1] Houde, S, Hill, C. (1997) What Do Prototypes Prototype?, in Handbook of Human Computer.
- [2] Alan Dix -Janet Finlay - Gregory Abowd - Russell Beal, Human-computer interaction.
- [3] Tanenbaum, Computer Networks.Tanebaum, Operating Systems Design and Implementation.
- [4] Datasheet PLCSR3101BFU
- [5] <http://www.can-ci.a.org/>