PC-Based Automation of a Multi-Mode Control for an Irrigation System

Azzouz Benzekri
University of Boumerdes, Algeria
Faculty of Engineering
Dept of Electrical Engineering
azizbenzekri2001@yahoo.com

Kamal Meghriche
University of Versailles S-Q, France
Versailles Lab of Sys Engineering, LISV
Mechatronic & Integrated Sys, MIS
meghrich@lisv.uvsq.fr

Larbi Refoufi
University of Boumerdes, Algeria
Faculty of Engineering
Dept of Electrical Engineering
L_refoufi@yahoo.com

***Abstract*--This paper describes the design and implementation of a low-cost multi-mode control for an irrigation system built around a personal computer (PC). The system uses in-situ soil water potential measurements, the weather condition parameters and the set points data provided by the user to decide when and how much water to apply to the irrigated field.**

The soil moisture content and the climatic parameters are monitored by a microprocessor-based data acquisition and distribution controller system under the supervision of the host computer.

A bi-directional serial link allows the host computer to receive, store and display in real-time the overall irrigation system status on the PC's screen. Also, the PC has the capability to transmit data to the controller and instruct it to operate in one of the three possible modes.

The firmware code written in assembly language and stored in an EEPROM along with a Delphi-based friendly graphical user interface was developed to manage irrigation and other related practices such as fertigation.

Full circuit and program codes were implemented to verify system operation.

***Keywords*-- Multi-mode control, PC-based, data acquisition, irrigation, soil moisture, serial communication.**

I. INTRODUCTION

The primary source of water in agricultural production in most parts of the world is rainfall. The three main factors that characterize rainfall are: amount, frequency and intensity, the values of which vary spatially and temporally. When the weather does not provide enough rainfall to feed agricultural needs, farmers should supplement water available through rainfall by some type of irrigation to manage the soil moisture and nutrient concentration to create the optimum-growing environment.

With limited availability of freshwater and increasing costs of energy and manpower, irrigation, which can contribute substantially to crop production should be planned and managed in such a way that no drop will be wasted.

Due to complexities in the precise knowledge of the rainfall's main characteristics, the irrigation scheduling cannot be planned neither on the minimum values of the average precipitation during the growing season nor on the maximum. The former may lead to an over-irrigation causing crop diseases and waste of water, fertilizer, and energy. Also, and besides running off and evaporating, the excess water will percolate deep to soil layers below the root zone with all nitrates and other pesticides it contains polluting the ground water. The latter on the other hand, leads to an under-irrigation causing a highly reduction in both crops quantity and quality. Therefore, deciding when to turn on the irrigation system and how much water to apply is a complex decision-making process.

The solution is the automation of the irrigation water management process based on sound scientific scheduling practices.

Actually, automated irrigations have been around since the mid of the 20^{th} century. The early controllers were basically simple timers and switches used to turn on the irrigation system for a predetermined period of time regardless of the weather conditions or the soil moisture content. Since that time, there was a continuous improvement of automated systems in all directions.

Most surface irrigations were converted to pressurized irrigations, while the early mechanical automation devices were continuously replaced by modern electronic controllers. In the last two decades, with the proliferation of powerful and low cost microprocessors and the impressive growth of PCs performances, mechanized irrigation harnessed the power of computerized controllers to improve water use [1]. This increase in the sophistication of automation rationalized the utilization of inputs, increased production, reduced losses and manpower yielding in an increase of the farmer's net income.

This paper presents the design and implementation of an automated PC-based multi-mode control for an irrigation system. The block diagram in Fig. 1 depicts the hardware organization for the implemented control system. The system consists of a PC (host computer) running Windows which uses Delphi-based software program to access and supervise the embedded microprocessor-based data acquisition system via the serial port.

The following elements are used in our irrigation control system. An in-situ controller built around the Z80 microprocessor with tranducing circuits and A/D converter, digital input/output (I/O) interfacing and a bi-directional serial communication with the host computer.

1-4244-0840-7/07/$20.00 ©2007 IEEE.

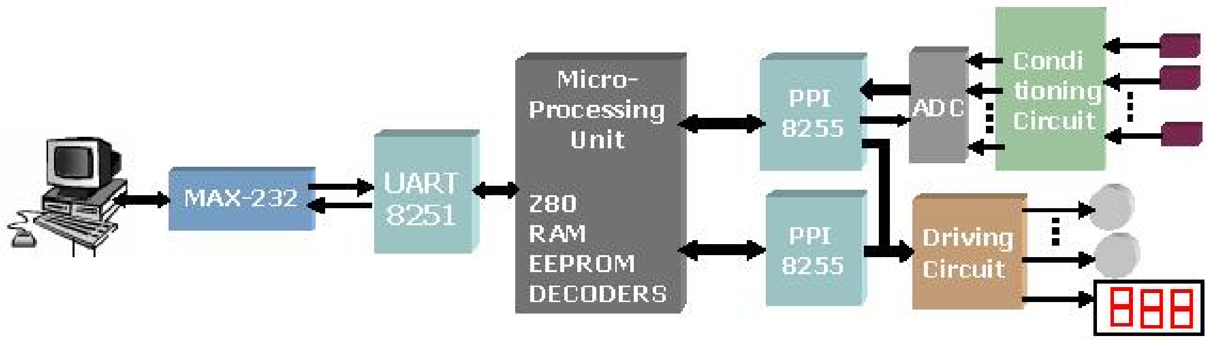

Fig. 1. Optimal irrigation control system block diagram.

The remainder of this paper is organized as follows. In section 2, we present a brief description of irrigation water management and the design approach. Section 3 discuses the development of the data aqcuisition system. Section 4 presents the flowcharts summarizing both the firmware and the software application programs of the system. The paper terminates with a conclusion which discusses several possible directions for continual research into the development of automated irrigation systems.

II. IRRIGATION WATER MANAGEMENT

Increasing energy costs and decreasing water supplies point out the need for better water management. Irrigation water management involves more than just turning on the irrigation system because it has not rained for a few days. Irrigation management is a complex decision-making process to determine when and how much water to apply to a growing crop to meet specific management objectives [2]. An optimum irrigation management needs rational and scientific methods particularly in regions where irrigation supplements rainfall and/or snowmelt. The irrigation management strategies should be designed to minimize water losses from evaporation, deep percolation and runoff. Also, an efficient irrigation management should not allow spraying during warm windy weathers. Blowing winds reduce irrigation effectiveness because they increase evaporation and affect the distribution patterns.

The runoff losses can be minimized or completely eliminated by applying water at a rate lees than or equal to the soil intake rate. The evaporation losses can be minimized when irrigation is conducted properly. Interrupting or delaying irrigation on hot windy days should be a common practice. It reduces significantly water losses due to evaporation and prevents low-uniformity of water application.

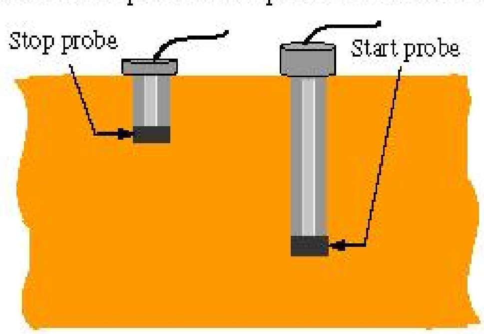

Fig. 2. Automatic moisture control using two probes: start and stop.

It is known that it takes some time for water to infiltrate through the soil; hence using a simple on-off controller would result in a waste of water past the root zone. By the time the water reaches the root zone, the system is triggered to stop irrigation, but the soil above the root zone being saturated; water will continue to flow below the root zone. One possible way to minimize deep percolation losses is the use of an anticipatory approach. This technique requires at least two soil moisture probes to properly monitor and control the water flow front, Fig. 2. One "deep" start probe is installed in the center of the root bowl of the plant and is set to the desired moisture level to indicate when irrigation should start. A second, stop probe installed below the surface anticipates the water movement by cutting off the irrigation as soon as it detects the advancing front flow. As soon as the upper layer is wetted to saturation, by gravity water begins to percolate to the root zone [3].

III. HARDWARE SYSTEM DESIGN

The core of the in-situ irrigation system controller is a Zilog's Z80 microprocessor. This 8-bit processor, running at a speed of 3 MHz provides a more than adequate processing power for this application [4]. A standard commercial PC could have been used, however, the inconvenience of having a PC in-situ with the necessary power supply can be considered excessive.

This embedded microprocessor-based data acquisition and distribution system is placed close to the point of measurements and actuators to minimize sensitivity to noise and transmission lines. Continuously, the system electrical variables (temperature, moisture ...) provided by various sensors are collected. These analog quantities are converted into numerical data by the analog-to-digital converter. The acquired data are then stored and displayed on the user interface. Furthermore, the controller performs some user defined functions (such as receiving data and commands from the supervisor host computer), performs some data processing to determine the irrigation time using an anticipatory approach, outputs commands to drive appropriate actuators, and handles the transmission of data and

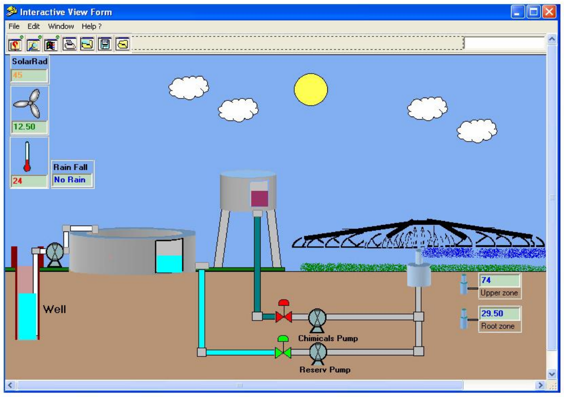

Fig. 3. The overall interactive irrigation system view.

irrigation system's status to the host computer for storage and display of the overall system on the PC's screen. Fig. 3 illustrates the overall interactive irrigation system view window at run time.

A. Development of the Data Acquisition Unit.

The data acquisition unit is built around the low-cost National Semiconductor 8-bit ADC0808. The device has 8 multiplexed analog channels out of which any one can be selected at one time [6]. When driven by a clock of 640 kHz, the ADC0808 performs a conversion rate per channel in 100 µs, a time conversion suitable for this application [5]. Fig. 4 depicts the schematic diagram of the data acquisition hardware circuit.

The ADC is interfaced with the microprocessor via a programmable peripheral interface, the Intel 8255A PPI. The 8255A PPI has three 8-bit ports. The input lines on the left which include the 8-bit data lines (D0-D7), two address lines (A0-A1), read/write (RD'/WR'), and reset (RESET) come directly from the microprocessor. The device is selected when the chip select (CS) input is asserted. This pin is driven by one of the outputs of the 74LS138 decoder. The controller uses the two least significant address bits A0 and A1 to access the internal data and control registers.

The analog-to-digital converter needs a clock to run. This signal is derived from the system's main clock. A binary counter, the 7493 is used to divide the main clock to suite the frequency required by the ADC0808. The address latch enable (ALE) and START inputs are connected together to simultaneously latch in the selected channel and start a conversion. The end of a conversion is signaled by the EOC signal. This output signal drives an edge-triggered D flip-flop, which in turn interrupts the microprocessor. After which, the microprocessor asserts the RD' signal for a read operation, before selecting the next input channel and starting the conversion.

B. The PC- embedded System Serial Interface Link.

It is the embedded system responsibility for sending the data and irrigation system status to the PC. Also, it receives set points data and commands to manage the irrigation system in either mode control through a serial digital communication full duplex RS232 standard interface. The unit is built around the Intel 8251 Universal Asynchronous Receiver Transmitter (UART), Fig. 5.

On one side, the UART is connected to the system's data bus and the synchronizing signals and on the other side, it is connected to the signal level translator, the MAX232, which in turn is connected to a DB9 connector [6].

To ensure synchronism between the host computer and the embedded system, the transmit and receive clock inputs are driven by the clock generated by a baud rate , the CD4040. the flip flop is used to divide the 3 MHz main frequency of the system to suite the UART clock frequency which should not exceed 1.6 MHz.

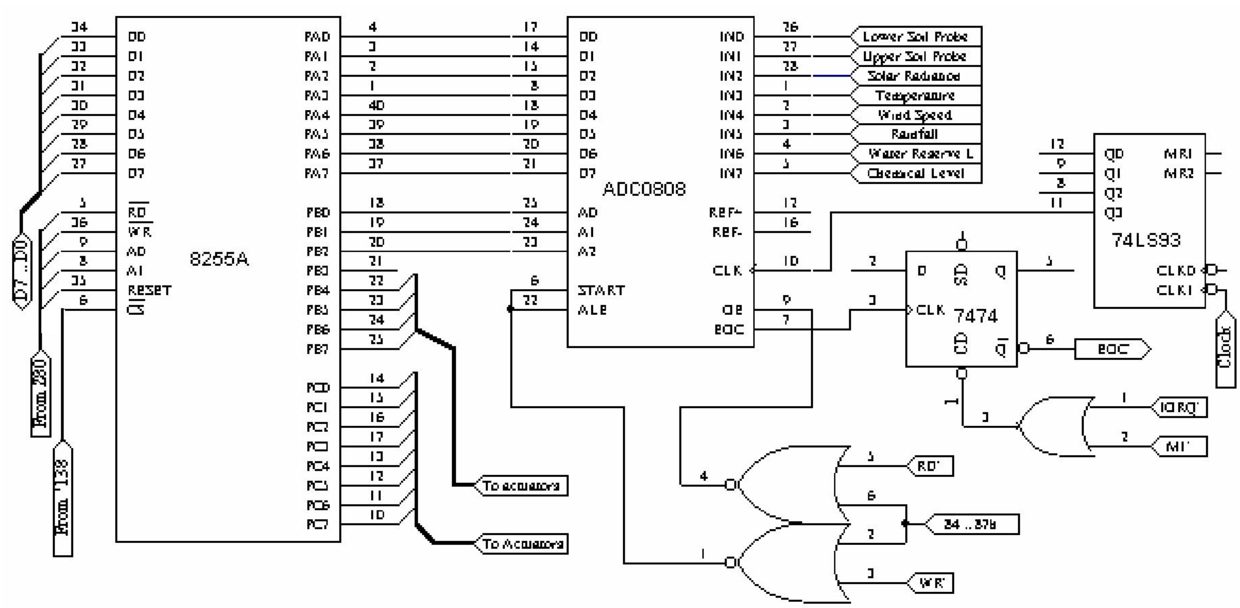

Fig. 4. The schematic layout of the data acquisition unit built around the 0808 ADC

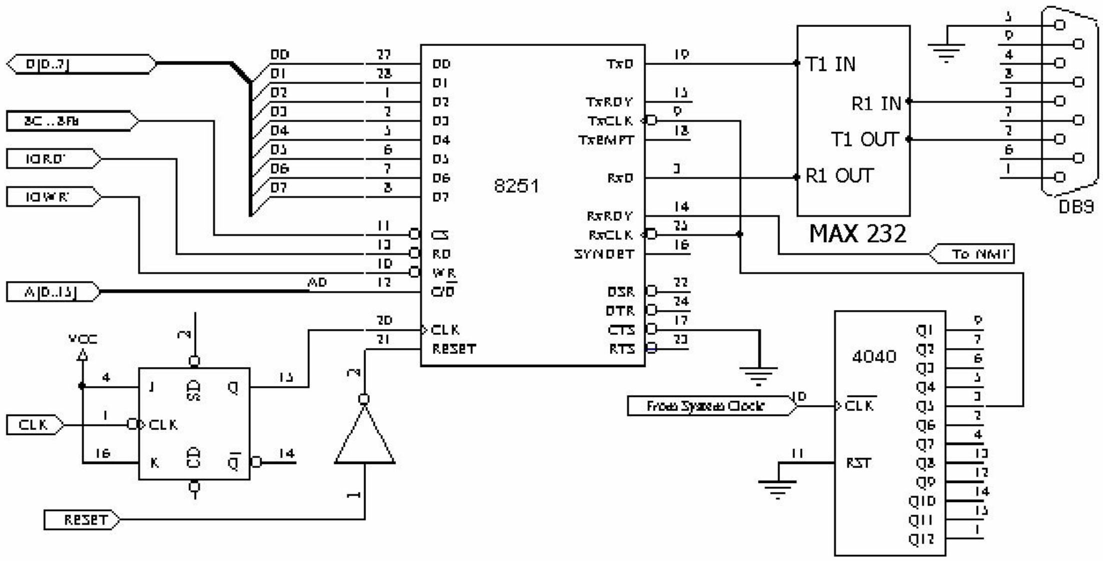

Fig. 5. The digital serial I/O communication interface with the PC's serial Port.

IV. DEVELOPMENT OF SOFTWARE/FIRMWARE

The system consists of a PC running Windows, which uses a Delphi based software program to interact with the embedded microprocessor-based system via the serial port. Taking into account all possible customer needs, the system is designed to operate in three independent modes: manual, semi-automatic or timer, and automatic. We developed a fully dedicated firmware / software program to acquire process, control and display the overall data and status of the automated irrigation system.

The general philosophy of the software system design was based on the following premises.

i- A friendly user interface

ii- Both software and firmware be based on a modular design basis.

A. The Firmware Program.

The firmware or (hard-coded program) stored in the EEPROM manages the embedded microprocessor-based system. It was written in assembly language, and then converted to machine language before being downloaded into an EEPROM for stand alone operation.

The program begins by initializing the hardware and establishing the link with the host computer in order to receive the set points data and control mode command. The flowchart of Fig. 6 summarizes the firmware program of the embedded system.

When the system is instructed to operate in the automatic mode, the data collection subroutine enables the system to acquire the data from the weather sensors and soil moisture probes. It selects the channel, starts the conversion, waits for the conversion to terminate, then reads and stores the data. This operation is repeated for all eight parameters. A processing routine reads these acquired data and the set points sent by the host computer and calculates if an irrigation process is to be initiated or not.

The acquired data and the irrigation system status are transmitted to the PC for display. Figures 7 and 8 show the window application at run time of the automatic mode and the corresponding flowchart. The window contains displays for showing the values and the current status of the different actuators.

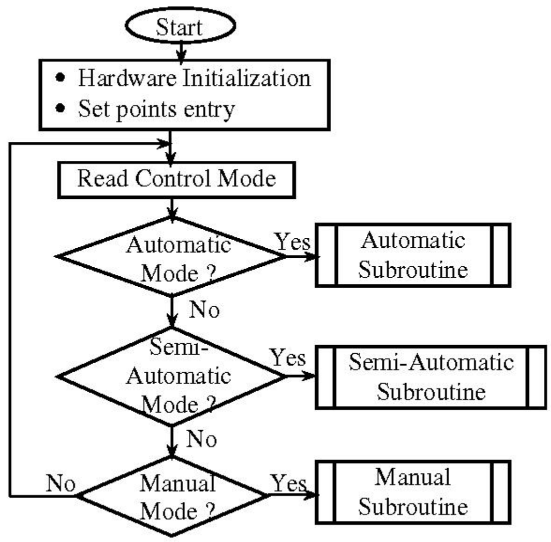

Fig. 6. The main program flowchart.

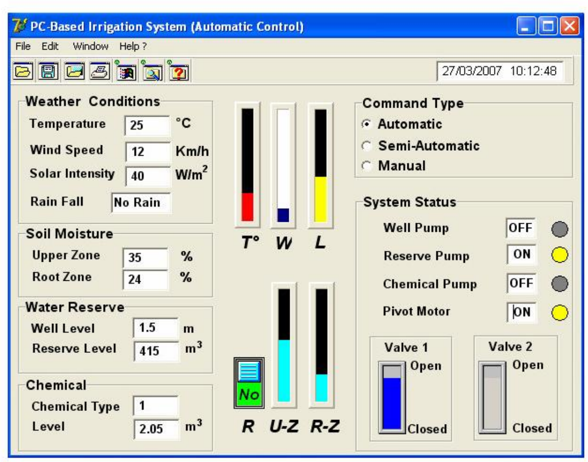

Fig. 7. The window application for automatic control mode.

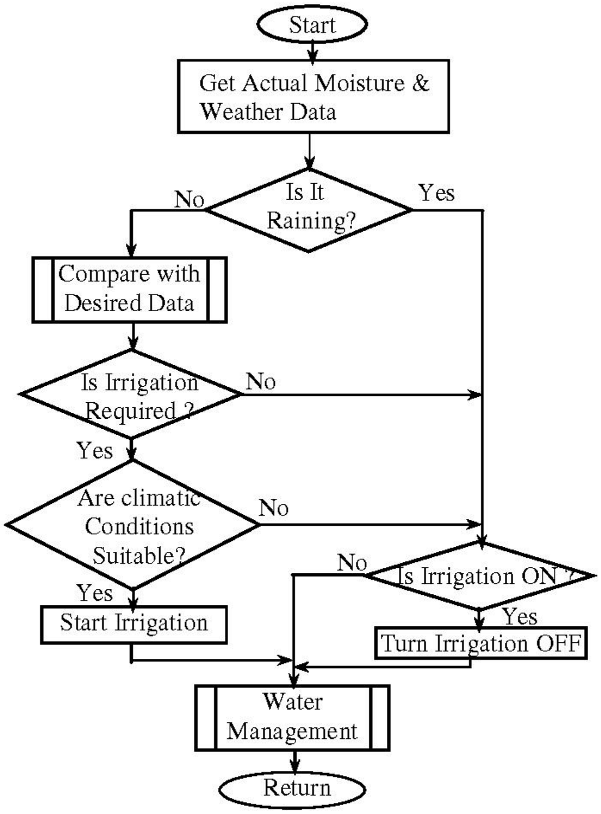

Fig. 8. The automatic mode flowchart.

In this manual mode, Fig. 9, the user has the freedom to activate the irrigation process irrelevant of the weather conditions and moisture of the soil. This feature may be of interest during the frost events, the irrigator can spray crops to warm up the atmosphere and thereby increasing the temperature.

In the semi-automatic or timer mode, Fig. 10, it is the farmer's duty to determine when and how much to irrigate. The irrigation application is conditioned however by certain climatic factors such as rainfall and wind speed. This mode may be used for chemical applications. In most irrigated agricultures, fertilizers and pesticides are generally applied with irrigation water. The graphical user interface allows the user to deliver the necessary fertilizer with great simplicity. The desired fertilizer solution is obtained by dosing the fertilizers controlled by valve-2 and the water flow controller by another valve-1.

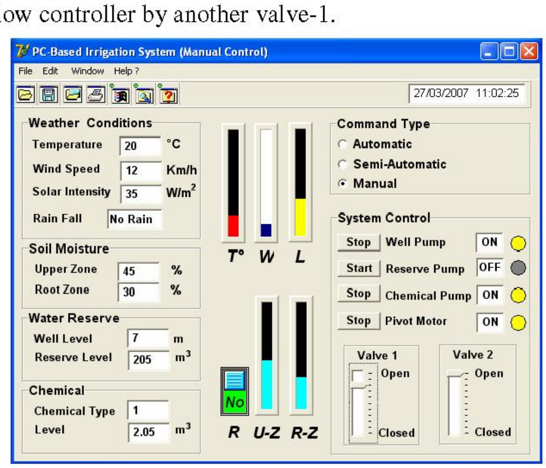

Fig. 9. The window application for manual control mode.

B. Development of the Graphical User Software.

The fully dedicated software program was written in Delphi to run under Windows XP platform. It has been written in such a way that the users get access easily to the irrigation system data, and have the ability to view in real-time the system's status. This developed friendly-user interface contains several windows, such the interactive view form, Fig. 3. This window displays graphically on the PC's screen a general view of the irrigation system. The window shows in real-time the operating status of the different equipments and the actual data of different parameters. With a simple click of the mouse on given equipment such as a valve or the tank will display a more detailed picture.

The manual, semi-automatic, and automatic interactive windows are illustrated by Fig. 7, 9, and 10 respectively. The RadioButton components in the command type GroupBox are used to select the mode of irrigation. It is possible for the user to change the mode while irrigation is on by simply enabling the corresponding RadioButton.

The software program has been developed to allow the user to specify the required target such as soil moisture levels, soil type, maximum wind speed to permit an irrigation to take place. Fig.11 illustrates the set points window.

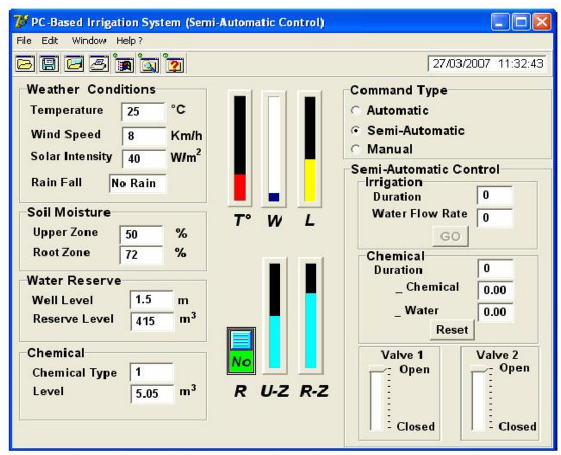

Fig. 10. The window application for semi-automatic control.

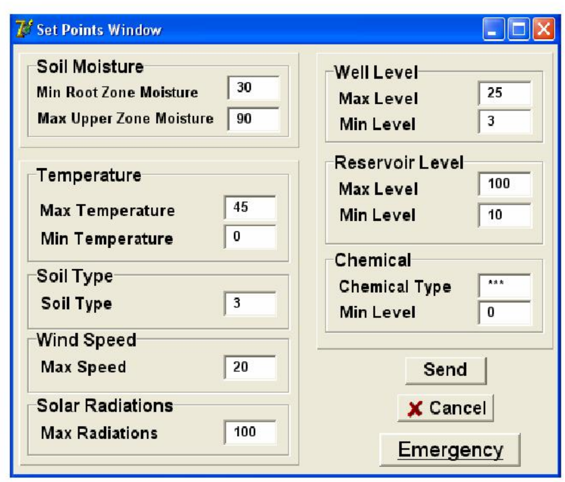

Fig. 11. The window application for data set points.

```
procedure Tform1.Button2Click(Sender : Tobject) ;
begin
   ApdComPort1.Open := true ;
   ApdComPort1.Output := char(strtoint(Edit6.Text)) ;
   Timer2.Enable := true ;
end ;

procedure Tform1.Timer2Timer(Sender :Tobject);
begin
   ApdComPort1.Output := char(strtoint(Edit7.Text)) ;
   Timer2.Enable := false;
end;

procedure Tform1.Button3Click(Sender : Tobject) ;
begin
    ApdComPort1.Open := true;
    ApdComPort1.Output := char(strtoint(Edit8.Text));
    Timer3.Enabled := true;
end ;

procedure Tform1.Timer3Timer(Sender :Tobject);
begin
```

```
  public
     { Public declarations }
  end;

var
  Form12: Tform12;

implementation

uses  Unit1, Unit2, Unit4, Unit17, Unit21;
{ $R  *.dfm }

procedure Tform12.TrackBar1Change(Sender: TObject)
begin
form1.trackbar1.position := trackbar1.position;
end;
procedure Tform12.TrackBar2Change(Sender: TObject)
begin
form1.trackbar2.position := trackbar2.position;
end;
```

Fig. 12. Fragments of the User Interface Delphi code.

Indicative fragments of the Delphi program code is given in Fig. 12. The first fragment code initializes the bi-directional serial communication between the PC and the embedded microprocessor-based system, whereas the second, illustrated by the second window controls the fertilizers and water flow valves.

V. CONCLUSION

An automated PC-based irrigation system which can be operated in several modes has been presented in this paper. The embedded microprocessor-based hardware with its firmware as well as the fully dedicated Delphi based graphical user interface were developed and successfully tested.

The fully dedicated graphical user interface of the system was developed with the premises to be easy to use and consistent. It depicts in real time the status of equipments, actuators, sensors, and controls. It was written in Delphi, a visual programming compiler that allows the programmer to program at a low level (assembly language).

We could have used a dedicated PC to monitor and control the irrigation, the inconvenience however, is that, having a PC in-situ with the necessary power supply and cables was felt excessive. A low-cost self-contained embedded microprocessor-based system supervised by a host computer was considered more attractive and can easily be fed by a battery or a solar panel.

The PC does not directly control the devices; instead, it manages the information flow and provides the user via the graphical user interface a means to remotely control the irrigation system by simple clicks of the mouse.

The developed system can be enhanced by some additional signal processing routines to display historical diagrams of the different data.

VI. REFERENCES

[1] F.S. Zazueta., A. G Smajstrla and G. A Clark, "Irrigation system controllers. IFAS Extension, Agricultural and Biological Engineering Department, Florida Cooperative Extension Service, University of Florida. Gainsville, FL. 2002.

[2] Black R. D and Rogers D. H, "Soil water measurements: an aid to irrigation water management". Available: http://www.oznet.ksu.edu..../L795.pdf.

[3] IIsystems, "Real time irrigation - a key feature of the Intelligent Irrigation System". Available: http://www.iisystem.com.au/HTML/strategy.html.

[4] R. S Gaonkar, *The Z80 microprocessor: Architecture, Interfacing, Programming, and Design*, Prentice Hall: 2001.

[5] H. J. Jeffrey, "Build your low cost data acquisition and display devices", Tab books, 1994.

[6] A. Benzekri and L. Refoufi, "Design and implementation of a microprocessor-based Interrupt-driven control for an irrigation system", in *Procedings of the 1st IEEE Inter Conf on E-Learning in Industrial Electronics.* Hammamat, Tunisia, 2006.